The handover and validation of as-built information according to Linked Open Data principles for digital asset management

A case study at Amsterdam Airport Schiphol

Anouk van Otterlo

Construction Management & Engineering
Eindhoven University of Technology

The handover and validation of as-built information according to Linked Open Data principles for digital asset management

A case study at Amsterdam Airport Schiphol

Author Anouk van Otterlo
Student number 0806204
E-mail a.v.otterlo@student.tue.nl

Date 15-03-2018
Date of final presentation 22-03-2018

University
Institute Eindhoven University of Technology (TU/e)
Faculty Faculty of the Built Environment
Master Track Construction Management & Engineering (CME)

In collaboration with
Company Amsterdam Airport Schiphol

Graduation committee
Chairman Prof. dr. ir. B. de Vries
1st Supervisor Ir. L.A.J. Mazairac
2nd Supervisor Dr. dipl.-ing J. Beetz

External advisors Ir. A. Worp (Schiphol)
Ir. M.H.L. van Bavel (BASED, BIM management & consultancy)

Technische Universiteit **Eindhoven** University of Technology

Preface

I am very proud to present this thesis as the result of my graduation project carried out in collaboration with the Eindhoven University of Technology and Amsterdam Airport Schiphol. After six months of hard working I finished my graduation project which will represent the end of my time as a student at my master Construction Management & Engineering.

During this time I can say that I learned, expanded my knowledge and grew as a person working on a topic which I am devoted to. The path I took answering my research question about transferring and validating BIM information to use and implement during digital asset management, turned out to be challenging, but interesting and never boring. While researching different topics about BIM, asset management and information exchange structures, I had to dig deeper in the data behind these techniques than I had imagined before I started. However, looking back I am happy I chose this path and I am glad I gained all this knowledge, which can help me during my professional career in the future.

While I developed myself during this time, I had help and support from different people around me. Without their guidance and efforts to push me further, I couldn't have reached this final point. First of all, I would like to thank my main supervisor Wiet Mazairac (TU/e) for his support, guidance and critical questions during the whole process which helped me to improve and get the best out of my work. In addition, I would like to thank Bauke de Vries (TU/e) for his overall supervision on the thesis and Jakob Beetz (TU/e) for his help with determining my research scope and giving technical support during the project.

Especially, I would like to thank my external supervisors Marcel van Bavel (BASED, BIM management & Consultancy) for his devotion, new ideas, valuable discussions and all his elaborated feedback during the project and Alexander Worp (Amsterdam Airport Schiphol) with providing me with insights in the organization, introducing me to all the departments and colleagues and helping me with getting a clear picture of the practical implementations of my thesis. In addition, I am very grateful for all the help of my colleague Maya Tryfona (Amsterdam Airport Schiphol) with resolving all programming issues I had. Without much experience in programming, I couldn't have managed this on my own.

Last but not least, I want to thank all my colleagues at Amsterdam Airport Schiphol, the BIM team and BASED, BIM management & consultancy for helping me during this period and for the great time and fun we had. I would like to thank all interviewees for taking the time and providing me with valuable insights for my research. Finally, I would like to thank my family and friends for all their help and support!

Anouk van Otterlo
Eindhoven, March 2018

Table of contents

Summary

Building Information Management (BIM) is a current innovation which is growing rapidly in the architecture, engineering, construction and owner-operated (AECO) industry. Even though BIM is often implemented into construction projects, its adoption within asset management is still lacking. The operations & maintenance phase is the longest and most expensive phase in the building lifecycle. The highest percentage of savings can be reached by optimizing processes, which shows the potential of implementing BIM in the future.

When BIM is implemented during asset management processes, not only costs will be saved, but also collaboration will be improved and processes will as well be more efficient. To achieve this, the information in BIM models should be of sufficient quality. If the information about all assets is complete, consistent and accurate after the handover of a project, the focus can be set on the desired performance of assets and real-time insights will be possible. The problem, however, is that currently no clear answer can be given to the question how as-built information should be transferred and validated for asset management. To answer this question, multiple problems are researched. First of all, the information in the BIM model should be clearly defined according to the requirements of the asset manager. This information includes graphical model information, non-graphical data attached to objects in the graphical model, and documentation. Besides this, the information exchange structure should be defined according to open standards and a validation process of this information should be established after the handover. The use of open standards will provide in a more practical and integrated solution, which can be adopted in the industry and not only by the specific company. When the information is transferred and validated, the information should be matched with the information structure of the asset management system. Finally, during this whole process the added value should be captured which will help to improve asset management processes in the future.

To answer these research problems, a literature study is firstly conducted. In this research, different open standards related to BIM, information exchange structures and asset management are elaborated. This study concluded that open BIM standards, such as IFC, IDM, IFD and BCF are recommended to be used during projects. However, the handover process to asset management is asking for a more elaborated information structure. According to the research the recommended structure which is future-proof and currently available is linked open data. This structure gives the opportunity to define the required information in an object-based database. To define the linked data structure an ontology can be used. An ontology can be seen as a kind of dictionary for computers, which defines all relevant entities, properties and relations between these entities and properties. Different ontologies already exist, such as IfcOWL and BOT, which can be used as a basepoint to align an User Defined Ontology (UDO) based on the requirements of the asset manager. To transfer information during the handover with the help of an information structure, the Information Container for Data Drops (ICDD) is proposed. The ICDD is based on semantics and can provide links between objects and related documents. Finally, validation of information should be an important step before implementing the new information structure into the asset management system, because without structured, uniform and validated information, no assured data knowledge, insights and added value can be generated.

Secondly, a case study is executed at Amsterdam Airport Schiphol, the organisation where the research is performed. In this case study the BIM ambition and vision of the company are presented, where after the different stakeholders and processes involved with BIM are

explained. Finally, different interviews are obtained with employees of the company, in which the results are discussed. Based on the case study, it can be concluded that though new processes are implemented, the stakeholders should be kept on board. The most important stakeholders involved with BIM in the company are the departments of asset management, project management and IT. Besides that, the requirements from the business perspective which should be taken into account during the implementation of BIM can be summarized in the following points; focus on both project management and asset management, state clear and concrete process steps, introduce direct implementable techniques and provide a clear and less complex information specification which can be adapted and validated. When a new database will be implemented this database should be object-based and compatible with all sort of assets and not only building assets.

The business requirements of the case study and the core data requirements identified in the literature study are put together in the methodology of this thesis. A complete overview is given of these requirements, together with the asset requirements. The asset requirements are divided into general asset requirements and specific requirements for an elevator. Based on the results of this overview, the new process model is initiated. This process model includes a handover container, which can be based on the ICDD standard. To use and process the semantics of this container in the asset management systems, a new environment should be set up, which is based on linked data. This new environment, or data hub, will be the link between the handover format and the management systems. To implement this process model, an UDO is modelled which contains the information structure and captures the connections between information in the data hub. When using this information structure, the new process will be object-based instead of document-based. The ontology is constructed according to the requirements and a basic structure that has already been established by the company. An alignment with other ontologies such as the IfcOWL and the BOT are made as well and the ontology can be easily extended with other assets, properties and functionalities.

The ontology is tested in a proof of concept. For this proof of concept an application is developed. The goal of the proof of concept is to demonstrate the usability of the ontology, to show that information from an IFC model can be converted into linked open data and that the information can be transferred and validated with the help of semantics. When using the application, an IFC file and a RDF file with an UDO should be loaded into the tool. After uploading the files the application will start checking if all IFC elements defined in the ontology are present in the model. When the objects are present in the model, these will be saved in a list. After saving the information in the list, the ontology will be parsed to an RDF graph. All elements from the IFC model will be added to this graph, including their attributes and properties. When the RDF file with the extracted information is instantiated, the data will be queried. The query searches for all defined elements, needed properties and attributes defined by the asset manager. The results of the query will be saved and exported in an excel file which generates a validation report. This validation report will state if all needed information is present and if the data is correct according to the datatype requirements.

The application is validated by testing 3 projects. Multiple IFC models are loaded in the application combined with the determined UDO. The results of the application are comparable with each other. It can be concluded that most information can be extracted from the IFC models but, still, some information is missing. Because of the linked open data structure however, there is a potential in using the application to find indirect relations between objects, properties and attributes with the help of queries.

The information structure based on linked open data is able to transfer, validate and use as-built information based on open standards and will thereby provide in an object-based structure. With the use of semantics, the as-built information can be validated automatically based on completeness and correctness of the data, however only a manual project-based validation is needed. When linked open data is used, the exchange of information between the different management systems will improve and can make new relations between applications or interfaces possible. When using linked open data for asset management processes in the future, it will give the best opportunity to capture more added value during the maintenance and operations processes.

Samenvatting

Building Information Management (BIM) is een huidige innovatie die hard groeit in de architectuur, bouw, constructie en beheer sector. BIM wordt veel geïmplementeerd tijdens de bouw, terwijl dit nog niet vaak geadopteerd wordt tijdens beheer & onderhoud. De beheer & onderhoudsfase is juist de langste en duurste fase tijdens de levenscyclus van een gebouw, waardoor hier het hoogste percentage aan kosten kan worden bespaard.

Het implementeren van BIM tijdens asset management zal niet alleen kosten besparen, maar ook samenwerkingen tussen stakeholders bevorderen en efficiëntere processen bewerkstelligen. Om dit te realiseren zal de informatie in de BIM modellen van een beoogde kwaliteit moeten zijn. Wanneer alle asset informatie duidelijk en accuraat is na de overdracht van een project, kan de focus worden gelegd op de benodigde prestatie van de assets en real-time inzichten geven in de informatie zal mogelijk worden. Het probleem is echter dat op dit moment geen duidelijk antwoord kan worden gegeven op de vraag hoe as-built informatie overgedragen en gevalideerd kan worden voor het gebruik bij beheer & onderhoud. Om hierin te voorzien komen de verschillende problemen die voordoen aan bod. Ten eerste zal de informatie in de BIM modellen duidelijk gedefinieerd moeten worden aan de hand van de behoefte van de asset manager. De informatie behoefte bevat een grafisch model met niet-grafische informatie gelinkt aan de objecten in het model en documenten. Hiernaast zal de informatie uitwisselingstructuur gedefinieerd moeten worden aan de hand van open standaarden, waarbij een validatie proces van de data vastgesteld moet worden. Wanneer de informatie gevalideerd is overgedragen, moet deze implementeerbaar zijn in de beheersystemen. Daarnaast zal de toegevoegde waarde tijdens dit nieuwe proces vastgesteld moeten worden om de asset management processen in de toekomst te kunnen verbeteren.

Een literatuurstudie is uitgevoerd, om in antwoord te kunnen voorzien op de genoemde onderzoeksvragen en problemen. In het onderzoek zijn verschillende open standaarden gerelateerd aan BIM, informatie uitwisselingstructuren en asset management uiteengezet. Vanuit de literatuurstudie kan geconcludeerd worden dat open BIM standaarden, zoals IFC, IDM en BCF aanbevolen worden om te implementeren tijdens de projecten, terwijl het overdrachtsproces vraagt om een meer uitgebreide informatiestructuur. Volgens de uitkomsten van het literatuuronderzoek is linked open data momenteel de meest aangeraden structuur om te implementeren. Deze structuur geeft de mogelijkheid om de benodigde informatie gebaseerd op de behoefte en eisen beter te definiëren. Daarnaast voorziet deze informatiestructuur in het initiëren van een object gebaseerde database. Een ontologie kan gebruikt worden om deze linked data structuur vast te stellen. Een ontologie kan beschreven worden als een digitaal woordenboek, dat alle relevante entiteiten, properties en relaties tussen deze entiteiten en properties vast legt. Verschillende ontologieën bestaan momenteel al, zoals IfcOWL en BOT, en kunnen gebruikt worden als een startpunt voor het implementeren en laten aansluiten van een eigen User Defined Ontology (UDO) gebaseerd op de eisen van de asset manager. De ICDD wordt voorgedragen als informatie container voor het uitwisselen van informatie tijdens de overdracht. De ICDD is gebaseerd op semantische informatie en kan informatie en documenten aan elkaar en objecten koppelen. Tenslotte is het belangrijk om een validatie proces te implementeren voordat de informatie structuur wordt gekoppeld aan het beheersysteem, want zonder gestructureerde, uniforme en gevalideerde informatie kan er geen garantie gegeven worden ten behoeve van de kwaliteit van de data.

Naast het literatuuronderzoek is er een case study uitgevoerd bij Amsterdam Airport Schiphol. De case study geeft de BIM ambitie en visie van het bedrijf weer, waarna de verschillende stakeholders en processen in het bedrijf met betrekking tot BIM worden beschreven. Tenslotte zijn verschillende interviews gehouden met werknemers bij het bedrijf waaruit conclusies getrokken zijn. Gebaseerd op deze case study kan geconcludeerd worden dat het belangrijk is om stakeholders betrokken te houden, wanneer nieuwe BIM processen geïmplementeerd worden. De meest belangrijke stakeholders betrokken bij de implementatie van BIM in het bedrijf zijn de afdelingen asset management, project management en IT. Daarnaast kunnen de behoeftes tijdens de implementatie van BIM vanuit het bedrijfsperspectief in de volgende punten worden samengevat; focus op project management én asset management, definieer duidelijke en concrete processtappen, introduceer direct implementeerbare technieken en voorzie in een duidelijke en minder complexe informatie leveringsspecificatie die geadopteerd en beter gevalideerd kan worden. Wanneer een nieuwe database geïmplementeerd wordt zal deze object gebaseerd moeten zijn en toegankelijk voor alle soorten assets, waaronder niet alleen bouwkundige assets.

De eisen vanuit het bedrijf gebaseerd op de resultaten van de case study en de data eisen gebaseerd op het literatuuronderzoek worden samengevoegd in de methodologie. Hierbij wordt een compleet overzicht van de eisen weergegeven waarin ook de eisen voor de assets zijn opgenomen. De asset eisen zijn opgedeeld in generieke asset eisen en specifieke eisen voor een lift. Gebaseerd op de resultaten in dit overzicht is een nieuwe proces model voorgesteld. Dit model bevat een overdrachtscontainer, zoals bijvoorbeeld de ICDD standaard. Voor het gebruiken en verwerken van de semantiek van deze container in de beheersystemen is een nieuwe omgeving nodig gebaseerd op linked data. Deze nieuwe omgeving, ofwel data hub, moet de koppeling worden tussen het overdrachtsformaat en de beheersystemen. Om het proces model te implementeren is een UDO opgesteld die de informatie structuur bevat en de connectie tussen de informatie in de data hub bewerkstelligd. Wanneer de nieuwe informatiestructuur wordt geïmplementeerd zal het nieuwe systeem object gebaseerd zijn in plaats van gebaseerd op documenten. De ontologie is gemodelleerd aan de hand van de eisen en de basis structuur in ontwikkeling bij het bedrijf. Hierbij zijn andere ontologieën, zoals IfcOWL en de BOT, gekoppeld aan de UDO. Daarnaast kan de UDO makkelijk uitgebreid worden door assets, properties of functies toe te voegen.

De ontologie is getest in een proof of concept, waarbij een applicatie is gemaakt. Het doel van deze proof of concept is het demonstreren van de toepasbaarheid van de ontologie, het aantonen van de conversie van de informatie in een IFC model naar linked open data en het bewerkstelligen van de overdracht en validatie van de informatie met behulp van semantiek. Wanneer de applicatie wordt gebruikt, moet er eerst een IFC bestand en een RDF bestand, met de UDO, worden ingeladen in de tool. Nadat deze bestanden zijn geüpload zal de applicatie starten met het checken van de gedefinieerde IFC elementen. Wanneer de elementen in het model aanwezig zijn, zullen deze worden opgeslagen in een lijst. Alle elementen uit de lijst worden toegevoegd aan de ontologie in de vorm van een RDF graph, inclusief de attributen en properties van de elementen. Wanneer de nieuwe RDF file is aangemaakt zal de data worden doorzocht. Dit wordt gedaan met behulp van een 'query'. Deze query doorzoekt de RDF graph op de gedefinieerde elementen en de benodigde properties en attributen. De resultaten van de query zullen hierna worden opgeslagen en geëxporteerd naar een Excel bestand. Dit Excel bestand genereert een validatierapport waarin wordt weergegeven of alle gevraagde informatie is gevonden, wat ontbreekt en welke informatie niet aan de eisen voldoet.

De applicatie is gevalideerd door middel van het testen van 3 projecten. Verschillende IFC modellen zijn hierbij in de applicatie geüpload samen met de UDO. De resultaten van de applicatie zijn vergelijkbaar met elkaar, waarbij er kan worden geconcludeerd dat de meeste informatie uit een IFC model kan worden afgeleid, maar dat er toch nog steeds informatie ontbreekt. De linked open data structuur geeft echter veel perspectief en mogelijkheden voor het vinden van indirecte relaties tussen objecten, properties en attributen met behulp van queries in de applicatie.

De informatie structuur gebaseerd op linked open data kan gebruikt worden om informatie over te dragen, te valideren en te gebruiken als as-built informatie gebaseerd op open standaarden waarbij een object-gebaseerde structuur wordt geïnitieerd. Met de hulp van semantiek kan de as-built informatie automatisch gevalideerd worden op compleetheid en correctheid, waarbij alleen nog een handmatige project inhoudelijke validatie nodig is. Het gebruik van linked open data kan het uitwisselen van informatie tussen verschillende beheersystemen verbeteren en daarbij nieuwe relaties tussen applicaties en interfaces mogelijk maken. Het gebruik van linked open data voor asset management geeft een goede mogelijkheid om meerwaarde te creëren tijden beheerprocessen in de toekomst.

Abstract

The current innovation Building Information Management (BIM) shows potential during the operations & maintenance phase of a building to save costs, improve collaboration and increase efficiency. After the handover, all the information in the transferred BIM models should be of sufficient quality and validated before being used during asset management. Open BIM standards, such as IFC, IDM and BCF are recommended to be used during project management, however the handover process asks for a more elaborated information structure. According to the literature research the recommended structure to implement is linked open data. When the handover information should be transferred, a container structure, such as ICDD, can be used to provide links between objects, documents and information. This structure also fulfills the requirements of the asset management derived from the case study at Amsterdam Airport Schiphol.

These findings result in a new process model, that initiates the ICDD as a handover format which will process the information to a new environment based on linked open data. Before the ICDD can be implemented, an ontology, which contains the information structure of the new environment, is defined. The UDO is constructed according to the defined requirements of the organization. An alignment with other ontologies and an easy extension of the ontology can be made. The constructed ontology is tested in a proof of concept according to a developed application. From the results of the application, it can be concluded that most information can be extracted and transferred from IFC models, even though, some information is missing. Because of the linked open data structure, however, there is a potential in using the application to find indirect relations between objects and properties with the help of queries. This new information structure is able to transfer, validate and use as-built information based on open standards and to provide in an object-based structure. With the use of semantics, the as-built information can be validated automatically based on completeness and correctness of the data. In the future, the use of linked open data will give the best opportunity to capture more added value during asset management processes.

List of figures

List of tables

List of Abbreviations

AECO	Architecture, Engineering, Construction and Owner-operated
AIM	Asset Information Management
ASM	Asset Management
BCF	BIM Collaboration Format
BEP	BIM Execution Plan
BIM	Building Information Management
BSDD	BuildingSMART Data Dictionairy
BSI	BuildingSMART International
CDE	Common Data Environment
COBie	Construction Operations Building information exchange
COINS	Constructieve Objecten en de Integratie van Processen en Systemen
EIR	Employers Information Requirements
GIS	Geographic Information System
ICDD	Information Container for Data Drops
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
IFD	International Framework for Dictionaries
MVD	Model View Definition
OTL	Object Type Library
OWL	Ontology Web Language
RDF	Resource Description Framework
SE	Systems Engineering
SPFF	STEP (Standard for the Exchange of Product Model Data) Physical File Format
UDO	User Defined Ontology
URI	Unique Resource Identifier
WWW	World Wide Web
W3C	World Wide Web Consortium

1. Introduction

The Architecture, Engineering, Construction and Owner-operated (AECO) industry is one of the main exploiters of natural resources. Because this industry creates almost all physical assets, which contribute to our own made society so many resources are exploited (Spence & Mulligan, 1995). The industry is scattered over many different disciplines and stakeholders and very conservative as well. Due to the nature of the industry, the adoption process of new innovations takes longer than in other industries, however such innovations can solve many problems, reduce waste and contribute to making more profit.

An innovation in this industry which is currently growing is called BIM. Building Information Management (BIM) is used more and more in the industry. According to the survey of the NBS (2017) 62% of the respondents say they are aware of BIM and already use BIM on some project, while this was 48% in 2016. Currently, 29% of the respondents say that they use BIM on more than 75% of their projects, while only 18% of the respondents use BIM on every project (NBS, 2017). The shift over the years is visualized in figure 1, which illustrates Rogers' innovation adoption curve. New companies starting with BIM were still part of the early majority until 2016, however they have become part of the late majority in 2017.

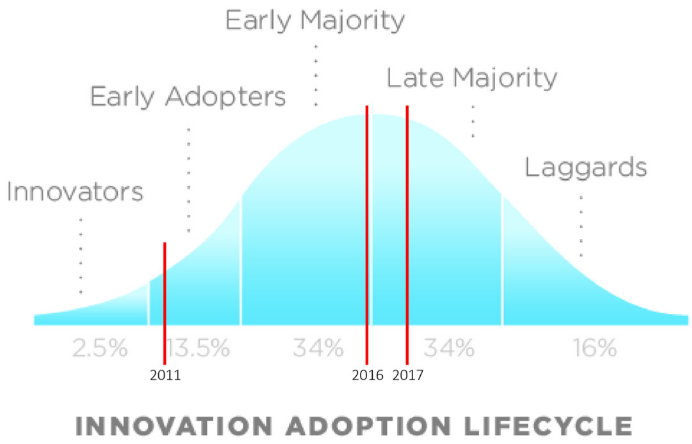


Figure 1: BIM adoption visualized in Rogers' standard innovation adoption curve

BIM is currently mostly used during the architecture, engineering and construction phase, however it can be implemented during the maintenance & operations phase of a buildings lifecycle as well. During the maintenance & operations phase of a building most costs are spent. However, during this phase the highest percentage of costs can be saved by optimizing processes using BIM (Ifma, 2013). Literature studies indicate the potentials of using BIM during asset management (C. Eastman, Teicholz, Sacks, & Liston, 2008; Ifma, 2013) and an interest is growing for using BIM during this phase. Currently, different open standards exist which can be used to implement BIM during the operations & maintenance phase. However, practical implementations are not developed fully to improve asset management processes.

1.1 Reading guide

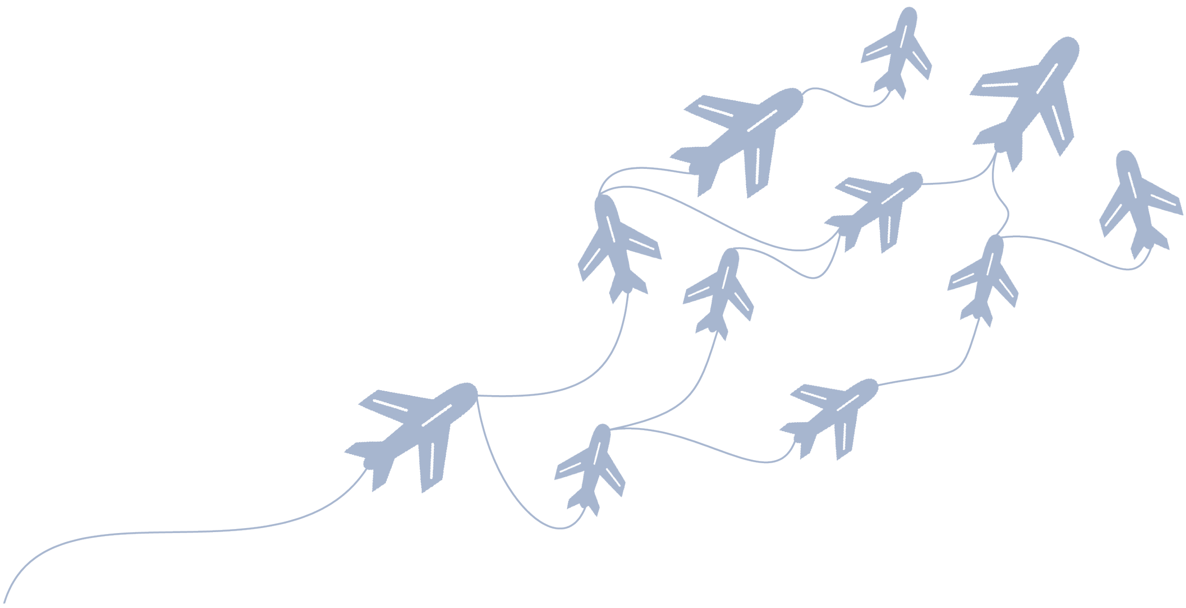
The report is organized in several main chapters, which are divided in sub-chapters. First, the introduction including a reading guide are elaborated in this chapter. The second chapter introduces the research approach. In this chapter the problem definition, research questions, research design and the importance of the thesis are described. Chapter 3 will elaborate the literature study, which focuses on the subject BIM and will provide a general overview about current practices and open standards in this field. Besides this, the literature study describes the information exchange process compliant with BIM practices and will give an overview of different information exchange structures. The chapter will conclude with describing asset management processes related to BIM. After the literature study, the case study is elaborated on. Chapter 4, will contribute to the findings at the company, Amsterdam Airport Schiphol, where the research is executed.

The literature study and the case study will give the basic findings of the research, where the next chapter will be based on. Chapter 5 contains the methodology, which is subdivided in multiple sub-chapters. Chapter 5.1 will focus on the determination of the requirements for exchanging information. Chapter 5.2 will provide the new information structure which is described as the model implementation and chapter 5.3 will give a proof of concept which explains the use cases and validation process of this new model. Lastly, chapter 5.4 will elaborate on the added business value after implementing the new information exchange process.

The thesis will conclude with chapter 6. This last chapter will present the conclusions and the recommendations for future researches.

2. Research approach

To give the outline of the report the research approach is described in this chapter. The approach is based on the problem definition and a certain gap in current research. This is described in section 2.1. To support the problem definition and define the scope of this research, the research question and sub-questions are elaborated on in section 2.2. In section 2.3 the design of the research is explained. Lastly, the methodological justification and the scientific importance of the research are indicated in section 2.4.



2.1 Problem definition

In the AECO industry the need of tracking and using information about the process of construction projects and maintenance is growing. Currently, high costs arise during the maintenance & operations phase of assets. The maintenance & operations phase is even the longest and most expensive phase of a building during its lifecycle (Becerik-Gerberet al., 2012). Using BIM during asset management can improve collaboration between stakeholders, reduce failure costs, improve the process and result in more efficient maintenance. BIM is used more in the industry. This growing ability to work with BIM can be seen mostly during construction projects, however BIM is still used scarcely during the operations phase of a building and limited practical examples can be found. Much BIM research has been done in several countries for over the past years and this increased from the year 2010. BIM is currently used in about 65 countries, and BIM is industry-wide adopted and researched. Areas which are most researched are; "process simulation and monitoring", "building information services" and "standardization". Operations & maintaining is a potential field to explore in the future (Badrinath, Chang, & Hsieh, 2016). The information extracted from the BIM models will have another purpose during operations & maintenance, compared to the purpose during construction projects. This will lead to different requirements of the BIM. For example, the non-graphical information about the assets will be most important during asset management, though 3D visualization will be less important. Before this information can be used during asset management, an important task is to validate the information on correctness. The handover of these models after the realization of the building will be an essential task.

Currently, it is not clearly described what should be transferred to a building owner after the realization phase. The information specification should be specified according to which information is needed and in what sort of format? Besides that, not all information is always needed in the model. This means it is important to filter the needed information for the asset manager. Secondly, it is still unknown how to transfer the information and how this information can be assimilated into asset management systems. When information about assets will be clear and accurate after the handover, a focus can be set on the desired performance of the assets during maintenance at the lowest possible costs. Through using correct information, real-time insights about assets will be possible and predictions can be made. This data can be derived from the BIM, after it will be validated. This will result in more efficient maintenance of the assets. A focus can be put on the risk and financial predictability.

To reach these goals more research should be done in this area. During the recent years BIM grew more important and many open standards were developed to improve the usability and operability in the sector. Because of the very scattered nature of the sector, it is important to use open standards and consistent developed processes. When this is standardized more, it will be easier to validate and control the output of the transferred information in the future. When validated information can be used, value can be added and captured during asset management.

2.2 Research questions

The current challenges described in section 2.1 has resulted in the following main research question: **"How to transfer and validate as-built information to capture added value for digital asset management?"**

To support this research question the following sub-questions have been identified:

- Which information does an asset manager need for digital maintenance and should be handed over after the construction phase?
- How should this information be handed over to the asset manager according to open standards?
- How can the asset manager validate this as-built information or how can the contractor suffice on the requirements set by the company?
- How can the company process this as-built information into their own asset management systems?
- What is the added value for asset management and how can this value be captured during this process?

The research problem related to the research questions is illustrated in figure 2. This figure shows the framework of the research. The main focus of the research will be on different specific assets. For these assets the information in the BIM should first be determined. So, what should be visible in the graphical model, which non-graphical data is attached to the model and if specific documentation is still required for these assets. This will answer the sub research question about which information is needed for digital maintenance. Secondly, how should this information be delivered to the asset manager and in which format, information container or information structure should the information be handed over and processed. The third step will answer the question about how this information can be validated. Should the information be checked with a validation tool or automatically? After the validation of the information the process of transferring the information into the asset management system should be researched. Which information is needed for which asset management system and should a filter be added to subtract the correct information for the specific system. Lastly, some attention will be given about how added value for the business can be captured during this whole process.

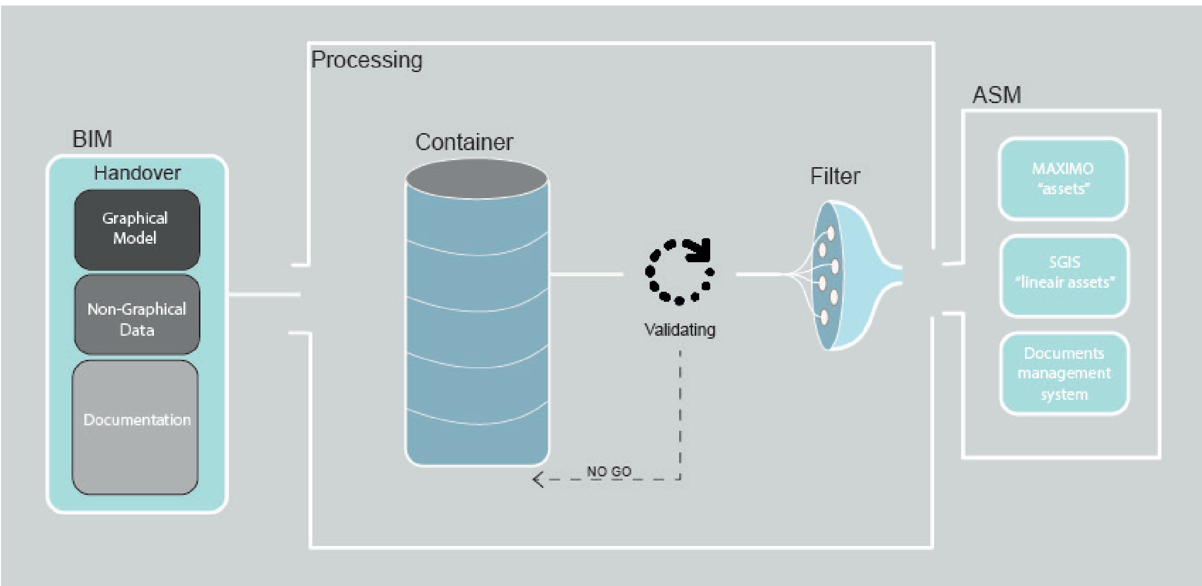


Figure 2: Research framework

2.3 Research design

This section will provide an overview of the methodological justification of the research and the research design divided in different phases. This research study will aim to find improvements for the asset management process, by using BIM. The case study will be executed at Schiphol Amsterdam Airport. Here for, a qualitative research will be obtained first. This research will include sketching the current situation at Schiphol. During this study the different departments involved in asset management will be taken into account. The current requirements set for these processes and the information needs of the asset manager will play a role. This information will be obtained by a quantitative research in the form of interviews. By interviewing different stakeholders at Schiphol, the different opinions and needs of the different departments can be reached. The goal of the interviews is to indicate how the building information set as required by the stakeholders, will be used during digital asset management.

Besides conducting the inventory research about Schiphol, the different currently existing open standards and data structures will be used to focus on in a literature study. The literature study will be focusing on the content of the BIM, the handover of the BIM and the validation of the BIM. In this research different existing tools, open standards and data structures will be investigated. An elaboration will be made on existing information, open standards and literature. The new findings derived from this research will be given and future implementation steps can be concluded. Using existing information and open standards, a more practical result can be reached. To support the results, a proof of concept should be made. To reach a more practical result, added value for Schiphol should be captured during the process and implementing this new process in the future.

The first part of the research will be the preparation phase. In this phase an elaboration is made on the initial research proposal. The literature study will focus on BIM related open standards and existing frameworks. The case study will give an inventory of the current situation, requirements and needs of Schiphol. The preparation phase will end with conclusions of both these studies.
The second phase of the research is the development phase. After making the conclusions of the preparation phase, recommendations are used to start working on the methodology. An information exchange structure will be chosen and will be developed further based on the requirements set by the user. After developing the information structure a proof of concept will be made and a validation of the information will be included. After making the proof of concept the possibilities are researched to transfer the information to the FM system. Finally, the new process structure will be reviewed and the added business value will be estimated.

The third and last phase of the research model is the reporting phase. In this reporting phase, the final conclusions of the whole research project are drawn.

2.4 The importance of the thesis

After finishing the thesis the results will be of importance from a scientifically, practically and social point of view. Regarding the scientific importance of the thesis, the thesis will contribute to and extend the existing literature available. It will further elaborate on using BIM for asset management. To complement this study open standards and open information structures are reviewed and considered academically.

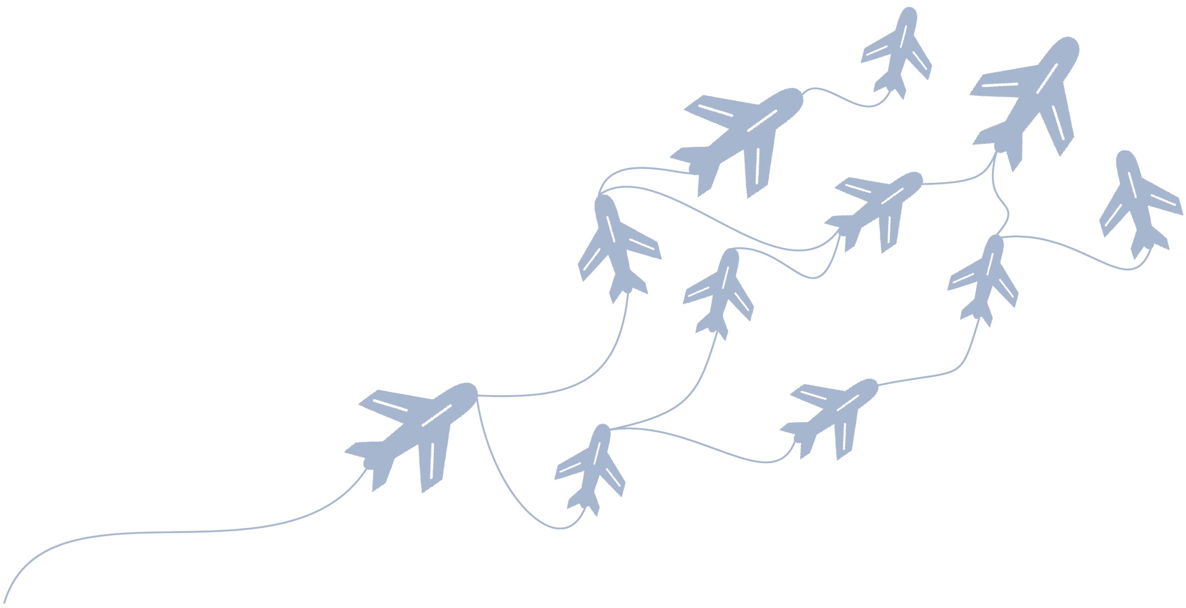
Considering the practical importance of the thesis, the proof of concept and implementation of the literature study will give a very useful and practical case study regarding using open standards for asset management. The insights of the thesis will provide for an improved working process which can be implemented in the future by the asset manager.

Socially, the thesis will be of importance as well. By using 3D asset information better in the future, a more efficient work process will be established. This will save costs and working hours in the industry in the future. Besides that, when projects are executed more digitally this will save in using and printing documents, which will be a more sustainable solution.

3. Literature Study

In the following chapter the literature study is elaborated. Different subjects which are relevant to the contents of this research will be introduced. One of the main purposes of the research proposal is to focus the research on open standards. Such standards will be used for the main structure of defining the model implementation. Open standards are important to improve collaboration. Because open standards are not dependent on software vendors, these standards can be used independently and will be developed further. This will enable the continuation of improving the process with the help of open standards in the future. The second purpose is to research and indicate which information exchange structure will be best to implement during asset management processes.

In this chapter a general introduction about BIM in the AECO industry is given in paragraph 3.1 first. Hereafter different open standards in relation to BIM which are supported by BuildingSMART are explained in chapter 3.2. This paragraph includes the open standards IFC , IDM, IFD, BCF and mvdXML. To express the importance of information exchange processes, this topic will be explained in paragraph 3.3. In chapter 3.4 different examples of information exchange structure are explained, such as COBie, Linked Open Data, semantic databases and information containers. To indicate the relation between BIM and asset management, this subject will be elaborated in paragraph 3.5. In this paragraph the importance of information for asset management and the validation of this information is explained. Chapter 3 will conclude with paragraph 3.6 and will explain the conclusion drawn from the literature study.



3.1 Building Information Management

BIM is implemented more and more in the Architecture, Engineering and Construction and Owner-operated (AECO) industry. BIM can be explained by the following definition of BuildingSMART:

"BIM is a business process for generating and leveraging building data to design, construct and operate the building during its lifecycle. BIM allows all stakeholders to have access to the same information at the same time through interoperability between technology platforms." (BuildingSMART, 2012).

The abbreviation BIM can have multiple meanings, namely; Building Information Model, Building Information Modelling and Building Information Management. For this research the concept Building Information Management is the most accurate, because BIM can be used during the whole lifecycle of a building. In the following chapters, the acronym BIM will be indicated as Building Information Management.

BIM is used to improve communication, management of information and collaboration during building projects (Sebastian & van Berlo, 2010). By focusing more on communication and efforts during the schematic design and design development phase, the efforts will be shifted to the beginning of the processes and this will lead to productivity improvement. By finishing the design earlier, more focus can be given to improving quality later in the process. This shift is illustrated in the MacLeamy curve in figure 3 and curve 4 illustrates a BIM process (Lu, Fung, Peng, Liang, & Rowlinson, 2015). Using BIM can increase efficiency and interoperability in the AECO industry. When BIM is used during design and construction projects it is important to make sure all information is correct. When the information is accurate and validated at the end of the project, this will create the possibility to use reliable information during the operations phase of the building.

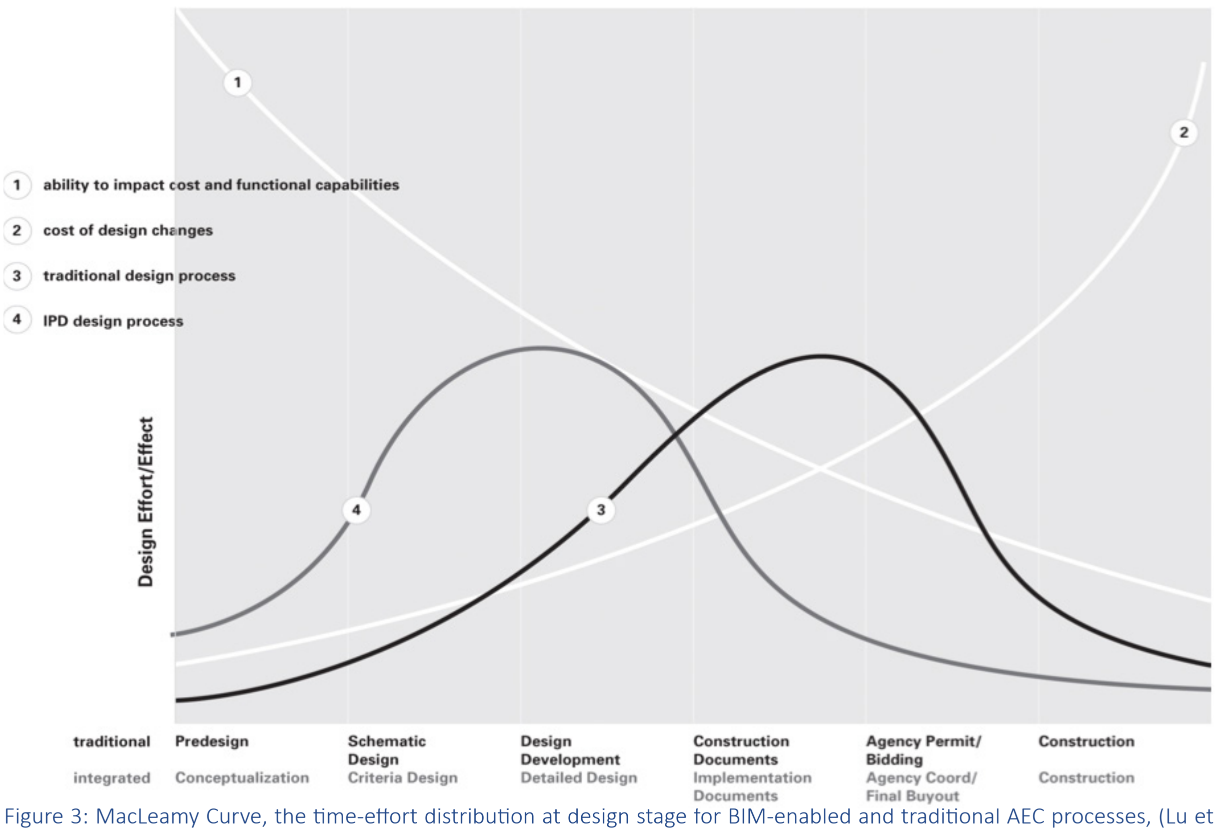


Figure 3: MacLeamy Curve, the time-effort distribution at design stage for BIM-enabled and traditional AEC processes, (Lu et al., 2015).

By implementing BIM in current working processes, different steps should be made. Before BIM is fully deployed, a distinction is made between the different levels of BIM maturity (Dakhil, Alshawi, & Underwood, 2015). The current state and quality of BIM can be indicated by these levels. The different BIM maturity levels are illustrated in figure 4. The figure indicates four BIM levels (0- 3). Level 0 is the base level of working in 2D and indicates the starting point of the growing model. The model has been set up by the UK BIM Taskforce and is adopted in Europe. Every level includes different data, tools, processes and working cultures. BIM level 1 is object oriented and involves working with 3D objects. In this level 2D objects are still possible. BIM level 2 involves file based collaboration. The object models of level 1 can be shared in a project. Everyone will work in his own model and the different models can be combined in one model. This method is called 'little BIM'. BIM level 3 will shift from file-based till object-based collaboration, which will provide for a more integrated level of collaboration. This method is called 'big BIM' (Bouw Informatie Raad, 2014).

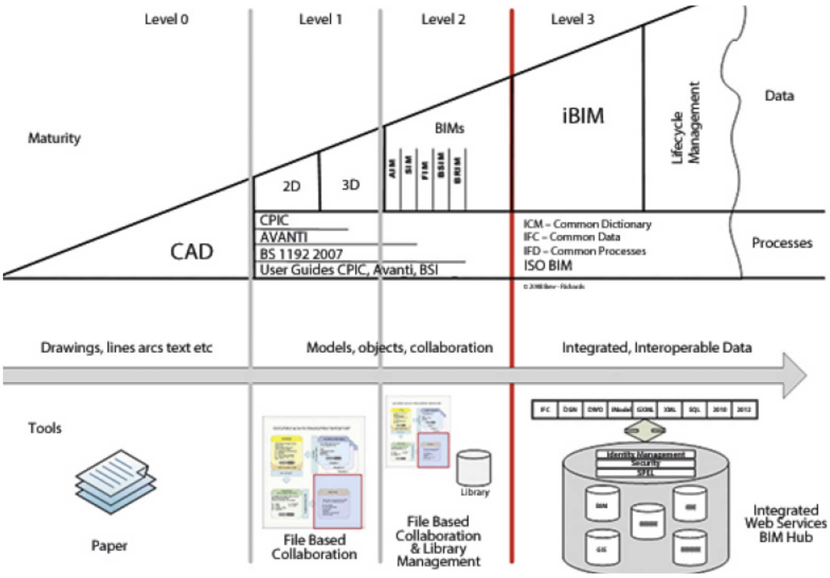


Figure 4: UK BIM maturity levels (Bew & Richards, 2008)

To optimize a building project, the concept 'Big Open BIM' should be used. When using Big Open BIM, the BIM models will be used by all stakeholders involved in the project with the intention to use throughout the whole lifecycle of the building. Besides that, the stakeholders will exchange files by using an open standard.

3.2 Open standards of BuildingSMART

In this chapter different open standards initiated by BuildingSMART are explained. IFC is the most important open standard of BuildingSMART and is explained in paragraph 3.2.1. Besides IFC, two other open standards exist, namely Information Delivery Manuals (IDM) and the International Framework for Dictionaries (IFD), see figure 5. These open standards are explained in paragraphs 3.2.2 and 3.2.3. BuildingSMART has also adopted two externally developed standards. First of all, the BIM Collaboration Format (BCF)is a standard, which can indicate and exchange clashes or errors in a

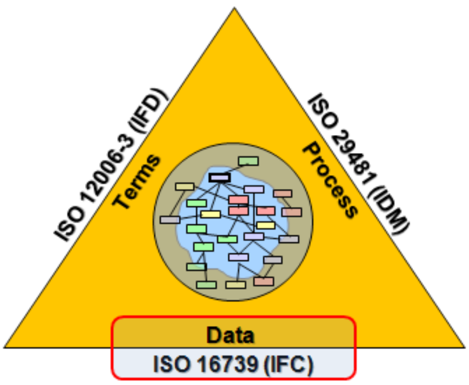


Figure 5: Overview of the BuildingSMART open standards, retrieved from http://www.buildingsmart-tech.org/

presentable format with the relevant stakeholders. Due to BCF is it easy to communicate between different stakeholders and solve issues. This is elaborated in paragraph 3.2.4. The second standards is the Model View Definition markup language (mvdXML), explained in paragraph 3.2.5.

3.2.1 Industry Foundation Classes (IFC)

The open standard IFC (Industry Foundation Classes) of BuildingSMART provides a standardized data format for the vendor-neutral exchange of digital building models (Borrmann, Beetz, Koch, & Liebich, 2015). Because IFC is an open standard, using IFC is obligated to reach Big Open BIM. An IFC file represents the BIM model of one or more particular disciplines of the project and can be easily shared with all stakeholders in a common data environment (CDE). Many software tools to design BIM models have adopted the IFC format and it is possible to import and export a 3D model to IFC with these tools. However, some limitations exist when importing and exporting IFC with this range of tools. Data loss occurs because of the differences between the IFC format and several software's data formats (Oh, Lee, Hong, & Jeong, 2015). To limit the errors and clashes in the IFC files, several software tools are available to view or check the model. Examples of such viewers are the BIM Server and Solibri Model Viewer or Solibri Model Checker. Different models can also be aligned in these software tools, which can make it easier to detect clashes between the different models. The clashes can be detected with rulesets and the mistakes found can be resolved by the responsible stakeholder (Berlo, 2010).

IFC can be exported and imported using the STEP Physical File Format (SPFF). In this format IFC is defined according to the EXPRESS schema (Pauwels, Zhang, & Lee, 2017). EXPRESS is an object-oriented modelling language, and applicable to building models. EXPRESS divides objects into classes, called entities. Entities can have attributes and relationships to other entities. Besides that, IFC can be translated to a form of XML, namely ifcXML. Using ifcXML the IFC format becomes more accessible and can also be expressed in formats, such as the XSD schema and the RDF graph (Borrmann, Beetz, Koch, & Liebich, 2015).

IFC includes five different classes which describe the contents of the file. These five classes are; functional type (objects, such as walls), geometry, attributes, relations between objects and behavioral rules (C. M. Eastman, Jeong, Sacks, & Kaner, 2010). Eastman et al. (2010) mention that *"a successful exchange can only be achieved when the exchange incorporates the semantically required subset of each of the aforementioned classes of description for each object in that exchange".* This indicates that all types modeled, should have a geometry, attributes such as properties and relations with other objects.

3.2.2 Information Delivery Manual (IDM)

An important factor during a project is to ensure that information can be retrieved and reused. IFC is mostly focused on data, however the process during a BIM collaboration will be important as well. When working with different stakeholders on a project, a streamlined process will be an important factor. In IFC many details can be specified which generates the possibility to make errors easily. To avoid this the specific contents, level of detail and the level of information of the model should be defined beforehand. To capture this information an Information Delivery Manual (IDM) can be used.

An IDM is a basic framework to develop and deliver a BIM model. The specifications of the IDM can only be determined if the purpose of the implementation of the BIM is clear. So the needed information is depended on the purpose of the user afterwards (C. M. Eastman et al., 2010). The IDM can capture and specify these processes and information flows. It will specify the relevance of the project, the stakeholders, which information should be created and how the information will be supported by software solutions (Building smart, 2010). Besides this, it enables the asset manager to ensure that the BIM meets the requirements set beforehand. Overall, the IDM will support collaboration and interaction about design decision consequences which will lead to a more efficient BIM process. The IDM is based on the ISO-norm CD 29481-2, IDM part 1 (Hoeber & Alsem, 2016).

3.2.3 International Framework for Dictionaries (IFD)

In a BIM model different objects are represented in a particular stage, with information attached to the objects. Because this information can be dynamic during the different stages of the project or can be inclusive in the model, a reference can be made to an open structure. This reference can be an Object Type Library (OTL) that stores dynamic information about all different objects (Hoeber & Alsem, 2016). Currently, different stakeholders in the building industry are working on such libraries which include building elements. These building elements are called 'concepts' which have properties and can be used in IFC. These libraries are regardless of languages, which means that an object is the same thing in different countries or languages. Besides this the library is compatible with other generic object libraries based on the ISO 12006-3 (Berlotti, 2012). An example of such a library is the BuildingSMART Data Dictionary (bSDD).

3.2.4 BIM Collaboration Format (BCF)

The BIM Collaboration Format (BCF) is an externally developed standard, adopted by BuildingSMART. It is an exchange format for clashes and errors in BIM models. A BCF uses the fileformat BCFzip, which is based on the XML schema. The current newest version of BCF is BCFv2.1. A BCF file represents a collection of issues of a model or between multiple models, which are linked to an element in the model by a GUID. The issues have a name, comments about the issue and a status. The issues can also be assigned to a specific stakeholder in the project. An open BCF-API exists to support exchange of the BCF file between different software solutions. Besides this, different communication platforms support BCF, where the issues can be uploaded and managed (Paasiala et al, 2017).

3.2.5 MvdXML

The second externally developed standard is mvdXML, which is based on Model View Definitions (MVD). According to BuildingSMART (2017) mvdXML defines a subset of the IFC schema in a Model View Definition to satisfy specified requirements. It includes validation rulesets as well. MvdXML can be devided in concepts which are dependent on rules. The concepts are described in a 'Concept Template' which is the basic structure and defines the root entity. The root entity should have predefined attribute values and referenced entities, which all can have additional constraints. The rules are defined according to this structure and will validate if this structure is provided in the model (Zhang, Beetz, & Weise, 2015).

3.3 Information exchange

Often there is a deficiency of information exchange between the contractor and asset manager, which will lead to inefficient information management during the lifecycle of a building (Hoeber & Alsem, 2016). The IFC standard can be used in a BIM process to exchange information between different stakeholders during a project. According to the British Standards Institution (BSI, 2014b) different steps have to be taken to reach a good BIM model or more specific; an AIM (Asset Information Model) for the maintenance and use of a building. To reach this, the

BIM model should be progressively developed starting at the initial brief stage until the build & commission phase which will end with a handover to the asset manager of the building, see figure 6. After every step in the process, an information exchange moment is indicated. These information exchanges include a graphical model, with attached non-graphical data. Besides that, there is still a need for documentation. This documentation includes for example specifications or schedules. The exchange of information should be implemented in a Common Data Environment (CDE). A CDE can provide a collaborative environment for all stakeholders in a project where all information can be shared (BSI, 2014b).

The different information exchange moments can be derived from the data drops systematics of COBie (Wood, 2012). Construction Operations Building information exchange (COBie) represents a framework to filter and sent data. This raw data will be transformed to information specifically needed for asset management and maintenance (Johnston, 2013). After each data drop a validation of the information in the data drop should be executed. After this validation the model can be improved during the next phase. Because of this process, at the end of the project the result can be better guaranteed.

To reach a good model, the asset manager should compose the Employer's Information Requirements (EIR). These requirements represent the needs of the asset manager, the time planning and coordination planning of the project. After describing the need, a BIM Execution Plan (BEP) should be drafted. This plan will include what should be done during the process and how the contractor or client should meet the requirements set. Lastly, the Master Information Delivery Plan should be made, which indicates how the information should be delivered to the asset manager (Bsi, 2013).

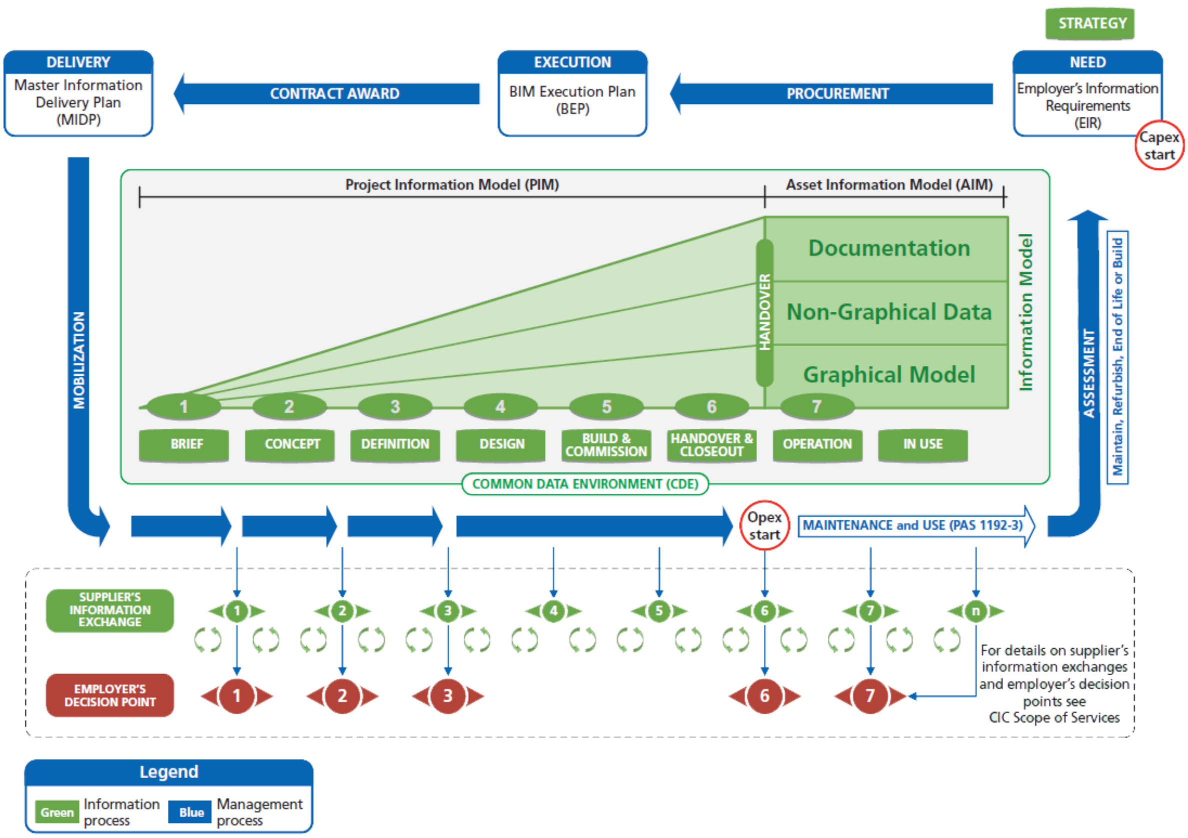


Figure 6: Information Delivery Cycle, PAS 1192-2 (Bsi, 2013)

3.4 Information exchange structures

Due to the rise of BIM in the AEC industry, a higher need to structure information will be demanded in the future. With the help of an IDM the specific need of information at the handover will be captured and the format or structure in which the AIM should be delivered will be specified. This information exchange structure will help with the handover of the correct and needed information in a familiar way to the asset manager. According to Eastman et al. (2008) information will be exchanged in one of the following ways;

- Direct, through links between tools, mostly with the help of application user interfaces
- Public product data model exchange formats which are based on open standards, such as IFC
- File exchange formats, which are not based on open standards
- XML-based exchange formats

Before this information exchange structure will be specified it is important to take the different sorts of data needed for asset management into account. These different sorts of data are extracted from different tools, databases or files. It is important to transfer data between these different data sources, which will request an open exchange format. When relationships between the different data sources can be made, the data from the BIM model can be connected between these sources. If a connection between facility management programs, GIS and other tools or databases exist, this will be of great added value. When different sorts of data should be integrated, some challenges will arise according to Curry et al. (2013);

1. Object identity and separate schema: data can be available in different sort of formats.
2. Data mismatch: data can be different if it comes from different sources.
3. Abstraction level: data is extracted from a particular viewpoint which means it can be incompatible with data from another abstraction level.
4. Data quality: the quality of the data is insufficient for its purpose.

Little interaction between different sources of information is possible, which results in these challenges (Curry et al., 2013). In the next paragraphs, multiple possibilities to link building data from different sources are explained. The different information structures can be classified according to a layer, indicated in figure 7. In this figure the different layers of linked information are illustrated. The first layer is the internet, which makes it possible to link information by linking computers. This means that information can be exchanged between different computers in a static form. The second layer is the World Wide Web (WWW), which will give the opportunity to link documents. Linking documents will give the opportunity to find the information on the web, but this is still unstructured. The third layer will provide for this structure in the information by the means of linked data. Lastly, the fourth layer, semantic web, will give knowledge and meaning to this structure (O'Keeffe, Alsem, Corbally, van Lanen, & Interlink, 2017).

Different information exchange frameworks already exist and can be classified according to the four layers of semantic levels. The frameworks selected for this literature research are already used or introduced in the AECO industry to improve the information exchange. The following frameworks will be explained according to their semantic level, usability and if these comply to open standards.

First, the framework COBie is explained, which is document based. Because the industry should be reaching BIM maturity level 3 in the coming years, more focus should be put on interlinked data networks (Rasmussen, Pauwels, Hviid, & Karlshøj, 2017). Linked open data will be explained where after an elaboration is made on semantic databases. To put more focus on the semantic web, Ontology Web Languages are discussed. The four ontologies IfcOWL, IfcWoD, COBieOWL and BOT are explained and compared. Lastly, the COINS container and the Information Containers for Data Drops are discussed.

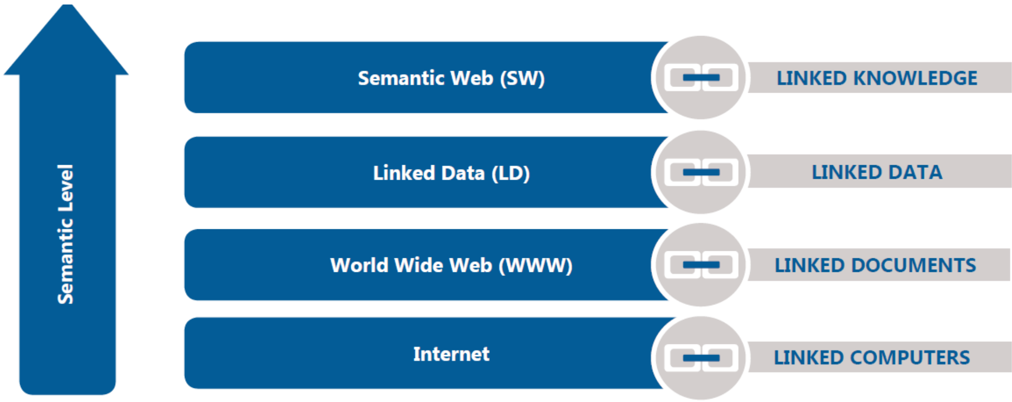


Figure 7: Four layers of semantic levels (O'Keeffe et al., 2017)

3.4.1 COBie

Construction Operations Building information exchange (COBie) represents a framework to filter and sent data. It is based on requirements and transforms raw data into information specifically needed for asset management and maintenance (Johnston, 2013). COBie focusses on the scope, technology, contract and process constraints of the facility (Ifma, 2013). COBie is an existing static and 'open source' format and is available in static formats such as spreadsheets or STEP. COBie generates one file for each building project. All phases of the building project are represented in COBie. The framework will represent design information, such as information related to spaces, zones, systems and equipment layout. The structure of COBie is illustrated in figure 8. Besides that, the framework will represent 'as-built' product data such as, as-built layout, serial numbers, warranties and spares (Tarcisio M Farias, Roxin, & Nicolle, 2015). It is possible to translate the information of an IFC file into a COBie spreadsheet file, using the IFC2x3 coordination view (East & Brodt, 2007). The information will be automatically translated and does not have to be translated manually. COBie has played a role in industry standards developed by BuildingSMART or the BSi (PAS 1192-4) (Patacas, Dawood, Vukovic, & Kassem, 2015).

The benefit of COBie is that the information represented can be used and generated without knowledge of the modelling programs of the initial BIM models and can be used for different sort of projects. Secondly, the framework can be easily validated on compliance, continuity and completeness (BSI, 2014a). Besides that COBie can be integrated in FM systems, so the asset manager can start the operations & maintenance phase efficiently (Love, Matthews, Simpson, Hill, & Olatunji, 2014). COBie locates all assets based on the space in the facility the asset is located in.

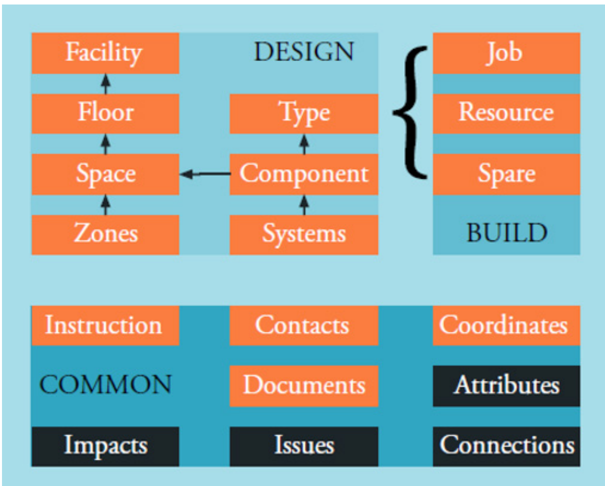


Figure 8: COBIe data structure, (Johnston, 2013)

The COBie deliverables can be implemented using the data drop method. The different information exchange moments as were mentioned in the information delivery cycle (figure 6, paragraph 3.3), can be derived from the data drops method of COBie (Wood, 2012). The five data drops indicate which data should be present at that certain stage of the project. Using this method can improve working towards progressively building a good BIM model. The final data drop includes all essential information to support asset management through the life cycle of a building. Finally, a total COBie deliverable is a complete set of operations and maintenance manuals (Tarcisio M Farias, Roxin, & Nicolle, 2015).

3.4.2 Linked open data

The second framework discussed is linked open data. To reach linked open data, linked data should be used as a starting point. As mentioned in the introduction of chapter 4, linked data elaborates on the concept of the world wide web. Linked data will give structure to all information on the web which is captivated in documents. Linked data is published data on the web, but which is not linked to outside data. This means that linked open data is linked data published under an open license. To retrieve good linked open data, the data should be compliant to the following rules, the five stars of linked open data according to Berners-Lee (2006);

★	available on the web but with an open license to be open data
★★	available as machine-readable structured data
★★★	available as plus non-proprietary format (e.g. CSV instead of excel)
★★★★	using open standards from W3C (RDF and SPARQL) to identify things
★★★★★	linked to other data to provide context as Linked Open Data

The different levels of the rules are shown in figure 9.

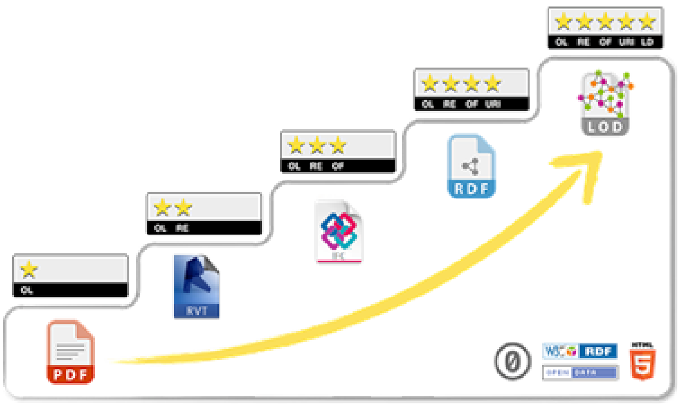


Figure 9: The five stars of linked open data

Linked data can be linked to provide context by the use of semantics, which will be the next step. Using linked open data or semantic web technologies can be used to capture added value for the stakeholders during the BIM process. Moreover, linked open data can be used to gain full semantic interoperability and to link information across diverse domains (Pauwels, Zhang, & Lee, 2017).

Linked open data can be applied to BIM. This means that (partial) models can be published as RDF datasets. In this case the GUID of an object is identified as an Uniform Resource Identifier (URI). The corresponding information of the object can be presented into RDF datasets (Törmä, Oraskari, & Hoang, 2012). These datasets will always be in the form of triples. This concept will be explained more in the following paragraph.

3.4.3 Semantic databases

The framework semantic databases will elaborate on the linked open data concept. This framework is described as the transformation of the web from a database of documents to a database that includes data and information which can be manipulated by a computer (Berners-Lee, Shadbolt, & Hall, 2006). Semantics are now commonly applied in different fields, including the AECO industry. Semantic web technologies are a means to represent information in structured graphs and make the information explicit. These structured graphs will provide meaning to the information (Pauwels, Zhang, & Lee, 2017). Such a structured graph exists of nodes and arcs. A node represents a concept or object and an arc will represent the logical relation between the nodes (Pauwels & Terkaj, 2016). These graphs are made explicit by formal structured and standardized knowledge representations (ontologies)(Sack, 2015). Semantics can be used to exchange and link different datasets.

A graph for the semantic web can be made with the help of the Resource Description Framework (RDF) format, which is based on Description Logic (DL) (Pauwels & Terkaj, 2016). DL can be described as a language which uses a formal knowledge representation. This language is used to describe concepts from a specific domain. According to Berners-Lee et al. (2006) the vision for RDF was to provide a minimalist knowledge representation for the Web. RDF graphs are directed labeled graphs and can be constructed with OWL concepts. RDF graphs have nodes which represents a concept or object, identified with a URI. Using the URI is necessary as a basic value to assign information and link to it. RDF graphs describe information with links between the different nodes, which makes information representable and reusable (Pauwels, Zhang, et al., 2017). The objects and links between them are based on logical statements and are named RDF triples, see figure 10 (Pauwels & Terkaj, 2016). The triple illustrates how two concepts can be connected. This way of connecting information, can be extended in an unlimited way. According to Berners-Lee et al. (2006) triple stores can have various purposes, namely;

- Providing a rich means to reason
- Storing large quantities of data
- Operating as plug-ins to current Web browsers
- Systems that can operate with a range of existing third-party databases

Figure 10: The triple form of an RDF statement: subject–predicate–object. (Pauwels & Terkaj, 2016)

It remains unclear how to define and manage semantic links between BIM models. If the exchange requirement can be specified, there will be an opportunity to make a semantic database and use this in the future. Using semantics can have multiple aims and opportunities. First of all, the interoperability in the sector can be increased. Semantics can help to improve the information exchange between different stakeholders. Secondly, semantic databases give the opportunity to link information across different domains or sectors, where for example different technologies can be linked such as BIM and GIS. So, semantic web technologies can increase the added value of BIM used for asset management in the field of interoperability and linking information from different domains which is based on a logical language. Such a logical language is OWL, which is a web ontology language (Pauwels, Zhang, et al., 2017).

3.4.4 Ontology Web Language (OWL)

The standard definition of an ontology by Gruber (1993) is a "formal specification of a shared conceptualization". In simple words, an ontology is a kind of dictionary for computers (Top, 2017). The ontology provides numbers and words with a meaning. An ontology is a data structure or formal naming which consists of all relevant entities, relations between entities and its properties and interrelationships of the particular domain depending on rules. A computer should interpret the ontology as a description of the reality. So through the semantic web a computer should deduce the meaning of the text and metadata and convert this. An ontology can be formalized by using Resource Description Framework (RDF) and Ontology Web Language (OWL). OWL will require some greater expressiveness between the object and relation descriptions than linked open data and because of this it is possible to refer to other ontologies in OWL (Berners-Lee, Shadbolt, & Hall, 2006). An OWL ontology is a RDF graph constructed with OWL concepts and syntax, which is based on triples. OWL further enhances RDF to make more complex RDF statements. The current used version of this ontology is OWL2 and uses the formal knowledge representation description logic (DL). (Pauwels, Zhang, & Lee, 2017).

An ontology is constructed in two parts, namely the Terminology Box (TBox) and the Assertion Box (ABox). The TBox expresses the general statements of the ontology. These statements are always understood to be true in the specific domain of this ontology. The ABox gives the instantiation of these statements. So, this means that these statements are assertions made, related to the statements in the TBox, at a certain point (Pandit, 2017). For example, the TBox can state that there will exist a building, which has building storeys. The ABox can indicate the certain building which is meant, for example the Schiphol building. The building storeys related to the Schiphol building, will be storey 0, 1, 2, 3 and 4. These particular storeys will also be stated in the ABox.

Using the RDF framework provides the possibility of querying the information captured in an ontology. A query will provide the information database with an assignment or action. If the query is defined correctly, information can be retrieved, created, updated or deleted. The query language for RDF is named SPARQL Protocol And RDF Query Language (SPARQL). SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions (Prud'hommeaux &Seaborne, 2008). A query can select information from a graph, update or add new information to a graph and delete information from a graph.

OWL and RDF are languages specified by the World Wide Web Consortium (W3C). The W3C puts effort in specifying, developing, and deploying languages for sharing meaning (Berners-Lee, Shadbolt, & Hall, 2006). Using these languages the two rules; minimum redundancy and keeping ontologies simple for easy maintenance are important to take into account for developing ontologies in the future (Rasmussen, Pauwels, Hviid, & Karlshøj, 2017). Many different ontologies have already been developed. A selection of these ontologies will be explained in the following paragraphs. The ontologies which will be explained and compared are; IfcOWL, IfcWOD, COBieOWL and BOT.

- **IfcOWL**

An OWL representation for construction data, which uses the IFC schema, is called IfcOWL. This ontology uses IFC to present the information as directed labelled graphs (Pauwels & Roxin, 2016). By making an ontology of the IFC schema, it can improve interoperability. As was mentioned in paragraph 3.1 Industry Foundation Classes, IFC is based on the EXPRESS schema.

EXPRESS is an object-oriented modelling language but lacks the formal semantics that is present in OWL and has a limited reuse and interoperability. However, EXPRESS can be transformed into a semantic structure, namely by creating a concept dictionary and classifying all concepts using relationships. For IfcOWL all entities in EXPRESS are defined as an owl:Class, which is a concept in the TBox of the IfcOWL ontology. All subtypes and supertypes are defined as relationships, which are indicated as rdfs:subClassOf of IfcOWL (Beetz, van Leeuwen, & de Vries, 2009).

To reach a good working IfcOWL ontology, different criteria have to be taken into account. The first criteria is that the ontology should be in the OWL2 DL format, to reach the right level of expressiveness. Secondly, IfcOWL should be matched as closely as possible to the EXPRESS schema and should be easily expressed into an RDF file (Pauwels & Terkaj, 2016). The consequence of this is that it will result in a very large and complex instance graph (Pauwels & Roxin, 2016). An option to reduce the complexity of IfcOWL is by using compartmentalization of parts of the EXPRESS schema (Beetz et al., 2009). This will lead to some extra steps and will show the disadvantages of the IfcOWL ontology. The advantage of IfcOWL is that it is based on the adapted and proved standard IFC and will probably be adapted earlier in the AECO industry in the future. Besides this, the ontology is specifically for this industry and can be aligned correctly with the modelled information models.

- **IfcWoD**

The second representation that uses the IFC schema is called IfcWoD and is based on the EXPRESS schema language as well. WoD stands for Web of Data. IfcWoD is also based on semantically modelling IFC relations in an ontology and used some basics of IfcOWL. This ontology differs from IfcOWL because reasoning capabilities are also taken into account. Because of this, the meaning of the entities, relationships, properties and attributes as defined by the IFC standard are put in the ontology. When using IfcOWL the considering relationship is mapped as an OWL class with an instance, instead of made with reasoning and OWL properties (Tarcisio Mendes De Farias, Roxin, & Nicolle, 2015). According to Farias et al. (2015) this has the following advantages;

- Simplifies and eases query writing
- Optimizes query execution
- Maximizes interference capabilities
- Allows for reducing data redundancy

The ontology is a semi-automatically method and transfers IFC into an ontology. IfcWoD uses a different approach than IfcOWL, but is still using some terms and definitions of IfcOwl. IfcWoD allows an easy application of the semantic principles (Tarcisio Mendes De Farias et al., 2015). The biggest advantage of this ontology compared to IfcOWL is the simplification of querying, but the ontology is still a very extended and complex representation (Rasmussen, Pauwels, Hviid, & Karlshøj, 2017).

- **COBieOWL**

The third ontology is generated from COBie instead of IFC itself. COBie is a static framework, which can be semantically enriched. This ontology is called COBieOWL and is directly made from the COBie spreadsheets data files based on the COBie 2.4 (Tarcisio M Farias, Roxin, & Nicolle, 2015). COBieOWL is designed to make a more interoperable standard in comparison to COBie, which is document-based. The difference between these ontologies is that IfcOWL is based on relationships between the several objects in IFC. In addition COBieOWL is based on the relationships between the indicated columns and cells available in the spreadsheets of COBie. Besides this, COBieOWL can be linked to other ontologies and can be extended as well.

COBieOWL is generated with the use of Java and an external Application Programming Interface (API) for handling spreadsheets. Each sheet in the COBie template, is mapped as an OWL class. Columns present in these sheets are mapped as OWL properties and cells on these sheets are mapped as property values, see figure 11 (Tarcisio M Farias et al., 2015).

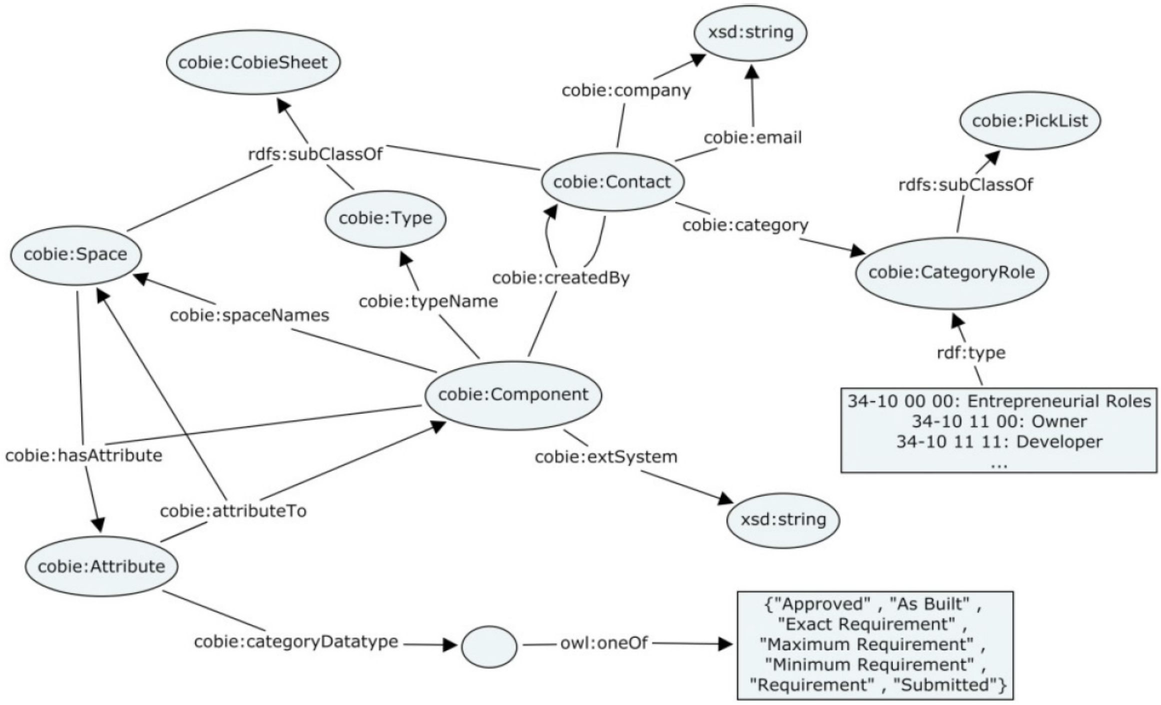


Figure 11: A portion of COBieOWL ontology (Tarcisio M Farias et al., 2015)

- **Building Topology Ontology**

Building Topology Ontology (BOT) is introduced by Rasmussen et al. (2017) as a reaction to all different topologies existing in the industry such as IfcOWL, which are very extensive and difficult to link. Because these ontologies are so extensive, they violate the rules specified by the W3C which were already indicated. By using a simpler ontology this can increase interoperability, reduce redundancies and can be used as an upper ontology. The BOT can serve as an ontology which is easy to maintain and can be extended or linked with other existing ontologies (Rasmussen, Pauwels, Hviid, & Karlshøj, 2017). According to Rasmussen et al. (2017) the core of the ontology exists of the following limited aspects:

- A building which is subdivided into storeys and spaces
- A space which can be bounded by building elements
- A space which can contain building elements
-

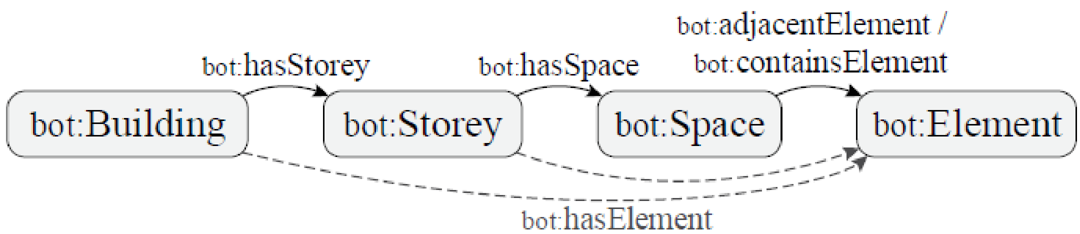


Figure 12: Simple Building Topology Ontology (BOT), (Rasmussen et al., 2017)

These aspects are illustrated in the ontology, see figure 12. This simple ontology exists of 4 classes and 5 object properties and is very minimal compared to other ontologies, but describes the essence of a building.

Lately, an improvement has been made on this ontology. The two classes Site and Zone are added. The class site was added because Facility Managers often have more buildings to maintain on one site. The class zone was added to support relationships between storeys and spaces (Pauwels, Schneider, et al., 2017).

3.4.5 COINS container

The COINS container stands for "Constructieve Objecten en de Integratie van Processen en Systemen" and is developed by a consortium of the building industry in the Netherlands which included various governmental authorities, contractors, educational and research institutes (Nederveen et al., 2010). The COINS container can be used as an exchange framework between users of BIM data. The container is a common Zip file with a predefined folder structure. This structure has two subfolders, namely one with the BIM model and one with all the other documents. COINS first structure has a predefined framework, which is difficult to adopt to specific needs and makes it sometimes difficult to work with. Therefor some improvements were made in the second version of COINS.

In 2017 the second version of COINS was launched, to improve the functionality and adaptability of the container. Because the first version was not based on semantics, semantic compatibility was not possible. IN COINS 2.0 there is an implementation of some of the semantic functionalities. COINS can be used as a semantic container and can use object type libraries (Willems, 2008). The framework is mostly document based and supports different types of documentation. The exchange framework is not object-based, so traditional documents still play an important role in this framework instead of the objects. However, the container has an open interface which makes it possible to link to software application through the open API. Each document in the container has an URI and it is possible to store all information history of these documents (Hoeber, Alsem, & Willems, 2015).

COINS has the benefit that it supports interoperability in the AECO industry, because COINS can be adopted by any discipline. To use COINS knowledge about programming or additional software is needed, such as the COINS navigator tool to create and edit COINS files (Hoeber et al., 2015). Because COINS is not totally understood and adopted in the industry, not many parties use COINS, and no user-friendly software is developed yet.

3.4.6 Information Containers for Data Drops (ICDD)

The ICDD standard is an exchange specification container, which is based on the COINS container. The basic data models of the two containers can be compared. The ICDD will be adopted as an ISO standard in the ISO 21597. The container format can store all sort of documents, such as BIM models, text documents, drawings etc. The container is also able to connect or link these documents. Text documents can for example be linked to the models. This basic structure of the container works in the same way as the basic structure of COINS. One of the differences between COINS and ICDD is that ICDD is separated in two parts; namely part 1 about the container and part 2 about the semantic dynamics. The separation is made, to make it easier to adopt the ICDD standard. The first part can namely be adopted without the second part, which makes it easier to implement.

The first part includes the structure of the information container, which is available in RDF/OWL. The information container includes an index, two different ontologies, directories to documents and information and the stored documents and information files (International Organization for Standardization, 2017a). The goal of the container is to provide a structure which enables to implement information which arises from the requirements in the process, delivery of 'as-built' information and specific functional purposes. The ICDD makes use of at least two ontologies which are linked together, namely;

- Container ontology
- Linkset ontology

The container ontology in the first directory gives the possibility to store all the different files in the container. The linkset ontology contains a data model which will map the relations between the documents in the first directory.

Besides the container format, the ICDD has a semantic layer added to the container which is also based on linked data principles. The semantic layer can be used by adding an extra ontology to the container which supports this goal. This will give the ICDD a more meaningful content and makes it possible to make links between different information available in the container (International Organization for Standardization, 2017b). Instead of only having links between the different documents in the container, links between objects in these documents can be made. By using the semantic layer in the ontology resource folders, user defined ontologies or user defined object type libraries (OTLs) can be added as well. This gives the opportunity to make a link between the container and the asset management database.

ICDD is currently still under development, so not many practical and easy-implementable solutions are ready to use. There is also no software available which will implement this technology now.

3.5 Asset management

During the operations and maintenance (O&M) phase of a building, risks and costs are relatively high. Asset management or facility management tries to maximize building functions, while ensuring occupants wellbeing. This purpose needs all sorts of information about these building functions and assets (Patacas, Dawood, Vukovic, & Kassem, 2015). The definition of an asset is stated in the ISO 55000 (2014) and gives the following explanation: "An asset is an item, thing or entity that has potential or actual value to an organization. The value will vary between different organizations and their stakeholders, and can be tangible or intangible, financial or non-financial."

All assets, or building components of assets have their different service live spans. Information about this service life data is required by the asset manager during the O&M phase of the building. The information about the assets will influence decision making processes in the development and execution of maintenance strategies. However, this is also dependent on the performance of the building components (Patacas, Dawood, Vukovic, & Kassem, 2015). Because service life spans are based on assumptions, the prediction of the life span is uncertain, which increases risks and costs during the building lifecycle. Besides monitoring the performance of a building, much more asset knowledge can be gained by using data optimally during asset management. To achieve this, the data should be of sufficient quality. According to O'Keeffe et al. (2017) information is of sufficient quality when it is complying with the following aspects:

- Be Complete
- Be Heterogeneous
- Be Consistent
- Be Reliable
- Have added value
- Be usable
- Be secure

These different aspects should be taken into account when using BIM for asset management. To control the data quality, validation of information is an important aspect. Currently, the information is still difficult to validate, but with the help of different software programs this can be improved in the future.

3.5.1 Asset management & BIM

Using BIM (Building Information Management) can decrease the risks and costs during the whole lifecycle of the building (Ifma, 2013). If BIM is used correctly, this has more benefits than only decreasing costs. Examples of such benefits are (C. Eastman, Teicholz, Sacks, & Liston, 2008);

- Increase building value by using analysis to increase building performances
- Assure program compliance against the requirements and different rules set
- Optimize asset management and maintenance by using the as-built model

Benefits of using BIM will arise for the asset manager when not only new technologies, but also new processes are managed. The organizations and people in it, should make changes at multiple levels. (Love et al., 2014). The handover of the new building with the needed documentation and information can be more streamlined and effective. After the handover, the BIM data can be used more efficient and effective and can exclude manual data entry in the FM systems. The data will thereafter be of higher quality and will also give many benefits during the life cycle and during enhancements of the building (Ifma, 2013).

BIM and asset management are a match because of the common need for data. BIM describes geometry, objects and attributes of an asset. It provides in non-graphical data attached to objects or assets. Also, relations between objects are possible in BIM. Moreover, the information, attributes and relations of assets are needed for asset management (Ifma, 2013). Still, asset managers should define their needed information requirements, to get the best data for their specific FM system. The requirements are not always clear to the asset manager and should be researched and defined better. A lack of use of open standards in FM can be a responsible factor for this unclearness and this is also seen as a barrier for improving the information handover (Patacas et al., 2015). Specifying the needs of the asset manager better beforehand will probably improve the results and willingness to adapt the BIM models before the handover.

3.5.2 Validating of information

When using and working with information, the validation of the information is an important part of the work process. Because all stakeholders in the AECO industry can work with the software of their own choice, the exchange of files should be with an open standard such as IFC. It is important to secure the quality of the BIM models in the exchanged files. BIM models can be checked quantitative and qualitative. A quantitative check will guarantee if all information needed is present in the model and a qualitative check will verify if all information in the model is correct according to the requirements. A distinction between how to verify if information is correct can be made as well. Information can be correct according to the prescribed data values, for example if a value should be a number, there should be a number and not a string. In addition a validation should be made if the value is exactly correct. So, if a number is filled in, this should be the correct number for the project. This check is based on the content of the project and is more difficult to automate. Finally, the quality and validation of the information in the model is also dependent on the implementations of the IFC export-import converter of the native software (Zhang, Beetz, & Weise, 2015).

Validation of the IFC models can be done manually or automatically. The checking of the models can be done with different software programs such as Solibri Model Checker or BIMserver.org. With the help of such programs, the models can be checked manually on visual aspects. Besides this, software programs are able to check parts of the IFC model automatically, so according to different self-enabled rulesets. Automated rule checking is defined as "assessing a design on the bases of the configuration of objects, their relations or attributes" (C. Eastman, Lee, Jeong, & Lee, 2009). According to Eastman et al. (2009) there are three different forms of rule representation, namely:

- Computer language encoded rules
- Parametric tables
- Language-driven; as a logic-based language or as a domain-oriented language

Based on the different rules, it is possible to use an automatic model view checker to validate a BIM model. An example of such a checker is the model view checker developed by Chi Zhang. This checker is developed based on the open standards mvdXML as the formats for structuring validation rules and the BIM Collaboration Format (BCF) to issue reports as a result of the checking (Zhang, Beetz, & Weise, 2015). The IFC files and the mvdXML are used as input for the checker and the BCF files will serve as the output.

3.6 Conclusions literature study

In the literature study different subjects relevant to the research about how to transfer and validate as-built information for digital asset management are introduced. The approach of this study was to focus on open standards. The literature study has shown that many open standards are available and are used to implement BIM in work processes. BIM is implemented more and more during projects and it can also be implemented during asset management processes. The literature mainly focuses on implementing the right processes and guarding the right quality of data information during maintenance and operations. If this is done in the correct way, communication, managing of information and collaboration can be improved. The use of open standards is highly recommended.

The open standards such as the file-format IFC, the formats IDM, IFD and BCF are good examples and can be used during projects to use and exchange the information. However, for the handover of information for asset management, a more elaborate information exchange structure should be used. It can be concluded from the information exchange structures which are researched that COBie has some limitations, because of the file-based structure of the format. The advantage of COBie is that it can be exchanged in an easy implementable format, which many software applications can import in a direct way. However, the information available in a COBie is not always complementary to the information needed during asset management. The COBie format only specifies a limited amount of information, in addition this can be different for every asset.

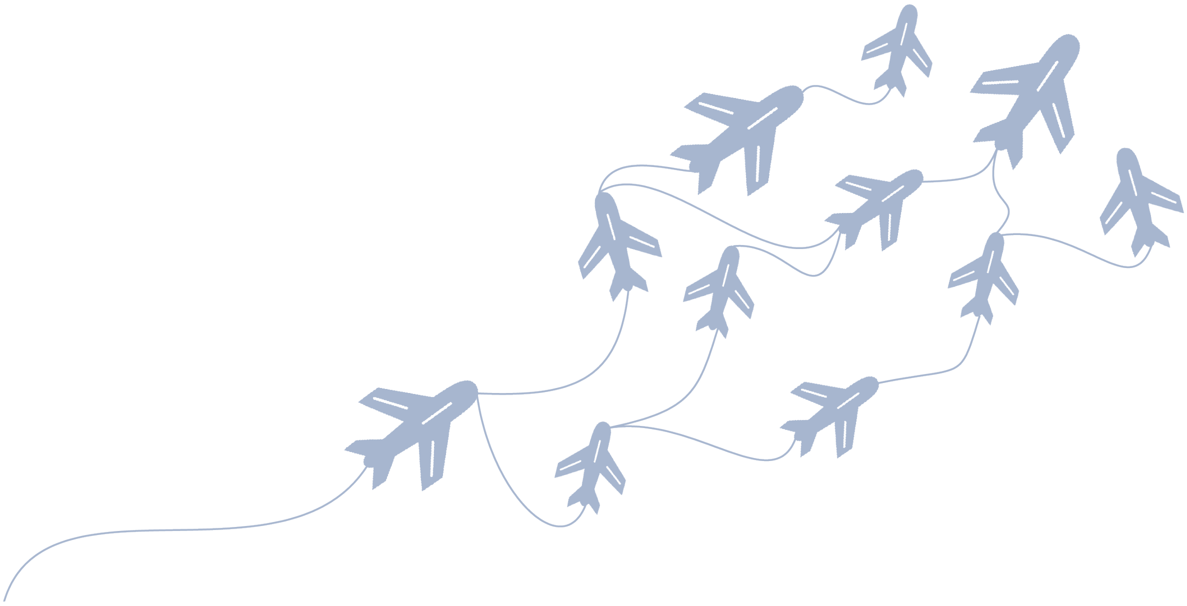
In comparison to COBie, linked open data or semantics will give more opportunities to influence the information according to the requirements by the asset manager. This can be done with the use of an ontology, which captivates the information structure of the needed information according to the requirements. The asset manager can influence the information structure per asset and dependent on the company. Besides this, linked data or semantics will be providing in a better solution because data can be linked object-based to different sources and is always based on open standards.

The different ontologies explained give a representation of the data structures from a particular viewpoint. IfcOWL and IfcWOD are both based on the EXPRESS schema of IFC. IfcWOD is easier to use, but still uses parts of the IfcOWL. Besides this, IfcOWL is more known and used more in the industry. COBieOWL is based on the COBie standard and focuses on maintenance information. COBie limits itself to the structure of the usual COBie file. In this ontology the sheets are represented as classes, and the cells on these sheets as property classes. This results in a more difficult aligning with other ontologies, which focus more on an object-based structure. Lastly, the BOT ontology is a very good example of a clear structured ontology, which only relies on the basic concepts of a building. This ontology can be used as a basepoint and be aligned with other ontologies. The best way to move forward with an ontology is to use parts of the BOT and IfcOWL ontology and align this with an UDO based on the requirements of the asset manager.

To transfer information during the handover, an information container can be used to link information to objects by using semantics. COINS is currently not developed and user-friendly enough. ICDD has a more clear structure and availability to use semantics and can be good to implement in the future. The information after the handover should be validated before being used during asset management. It is important to guarantee completeness, heterogeneous, consistency, reliability, added value, usability and security of the information.

4. Case study at Amsterdam Airport Schiphol

After the literature research, a case study is executed. The case study described in chapter 4 includes an inventory of the current situation at Amsterdam Airport Schiphol, the company where this research is obtained. The current state of the company and the BIM ambition and vision of Schiphol concerning BIM is described in paragraph 4.1. In paragraph 4.2 the internal and external stakeholders and the structure of the company are described. A link is made between these stakeholders and the current structures and processes regarding to BIM in paragraph 4.3. Lastly, different interviews were conducted which are summarized in paragraph 4.4. Furthermore, a conclusion of the case study is given in paragraph 4.5.



4.1 BIM ambition at Schiphol

Amsterdam Airport Schiphol is the biggest airport in the Netherlands and an important airport of Europe. Schiphol connects the Netherlands with the rest of the world. In the year 2016 Schiphol transferred a total amount of 63.6 million passengers and this number is growing every year (Schiphol Group, 2017). An airport is a complex organization. It has to fulfill the role of an owner, operator and maintainer at once and has different important jobs as well. Examples of the most important jobs are;

- Land aircraft
- Facilitate passengers, cargo and baggage
- Make profit
- Maintenance and asset management

To facilitate all the daily activities at Schiphol, assets are necessary. Every year new assets are added and old assets are replaced. Amsterdam Airport Schiphol (AAS) aims to have “best-in-class asset management”. Operations are a continuous process at an airport, however there is an information lack about these operations at the same time. To gather and store this information can have many gains in the future. By using (existing) information better, an airport can have more real time data and be more predictable. The focus is set on the desired performance of the assets of ASM at the lowest possible costs. Through using validated information, asset information can be kept up-to-date and real-time insights will be possible. This data can be extracted from the BIM and will result in more efficient maintenance of the assets. Schiphol will focus on the risk and financial predictability.

At Amsterdam Airport Schiphol an ambition concerning BIM has been composed. The ambition of Schiphol is to use open BIM to reach best in class asset management. The BIM ambition can be seen in figure 13. Schiphol wants to facilitate grow, but cannot expand their current area. Moreover, they want to position themselves as the ‘Best Digital Airport’ by managing a ‘digital twin’ of all the asset information during the asset lifecycle. This will optimize their process and will facilitate grow according to performance. Secondly, they want to improve their image by reaching more scheduling stability to make sure deadlines will be made. More stability can speed up the processes, reduce costs, improve customer satisfaction and add extra functionality.

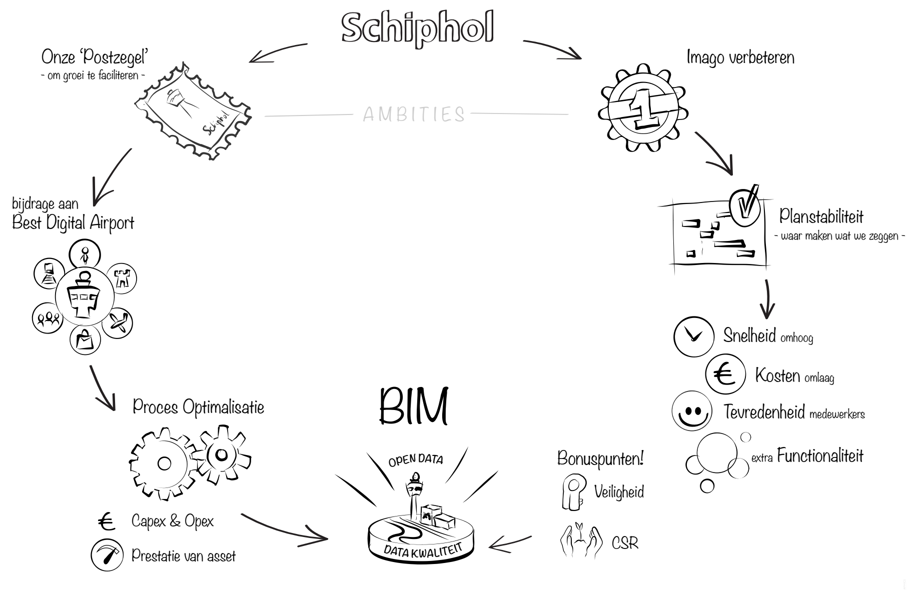


Figure 13: BIM ambition by Schiphol (2018)

To support the ambition of Schiphol three strategic key points are formulated where BIM will have an important role in the future of the airport:

- **Optimal asset efficiency (rendement)**: the assets of Schiphol should perform with less risks during the total lifecycle of the assets for the lowest costs.
- **Excellent organization:** at Schiphol integral thinking, co-working and taking responsibility are important to reach a good BIM culture among the organization.
- **Innovativeness & digital:** Asset Management Schiphol will create added value by actively focusing on innovation management for their assets, processes and digital developments.

4.1.1 BIM vision

The BIM vision of Schiphol is connected with reaching the maturity levels of BIM, explained in paragraph 3.1. These maturity levels will among other things use the concepts of 3D till 7D to indicate the specific elements and benefits of using the BIM models. According to the maturity levels, Schiphol formulated to go from 2D until 7D in 2027. Schiphol estimated that their organization is currently working on BIM level 1 and is making the transition to 3D. During the coming year Schiphol would like to extent their knowledge and processes until BIM level 2 and aim to operate their projects in 3D at the end of 2018.

3D covers the BIM model, which makes model walkthroughs, clash detection, project visualization, virtual mock-up models and prefabrication possible. Besides working in 3D they will focus on the next levels of dimensions. 4D will focus on the time, where construction planning and management and schedule visualization will be included in the BIM. This will be important for Schiphol, because Schiphol facilitates multiple projects which overlap each other, and also overlap with 24 hour open facilities. 5D will add the aspects of costs to the model, such as quantity takeoffs and ´real time´ cost estimating. 6D will focus on maintenance, which includes lifecycle management, data capture and sustainability. 7D will focus on an overall facility management application. In the end Schiphol aims to be compliant with BIM level 3 in the year 2027, see figure 14.

Implementing the different aspects in the BIM process will give more added value to the asset management process at Schiphol and more benefits will be gained during the whole life cycle of the assets. This means that the three strategic key points can be reached over time.

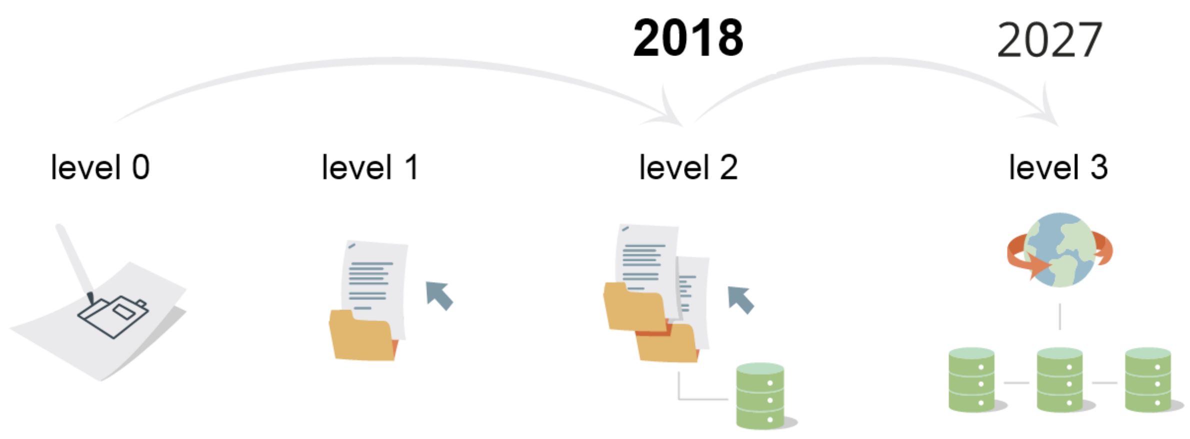


Figure 14: Goals for reaching the BIM levels at Schiphol Airport

4.2 Stakeholders

An important part of implementing BIM is getting all stakeholders on board. Implementing BIM can succeed or fail because of the attitude of management and the organization. In addition, BIM will change the business and it will change internal processes and daily tasks of employees. BIM provides in a better integral way of collaborating which will lead to a more agile way of working together. Amsterdam Airport Schiphol is a large organization. Because of the size of the organization, many different stakeholders should be taken into account. The different stakeholders for implementing BIM at Schiphol can be divided in two groups, namely internal and external stakeholders. The external stakeholders are mostly the (main) contractors and advisors Schiphol is working with during building projects.

The internal stakeholders at Schiphol which will be involved in the building projects and the AECO industry are illustrated in figure 15. Six different main divisions are indicated at Schiphol Group. First of all, the division 'Corporate Procurement' is responsible for the procurement at Schiphol. Secondly, the division 'Capital Programme' is indicated, which focusses on project management for all businesses around the "A-gebied", a particular area at Schiphol. In addition, the division 'PLUS' (Projecten Luchthaven Schiphol) focusses on project management at all other areas at Schiphol. Fourthly, the division 'SRE' (Schiphol Real Estate) is responsible for all commercial buildings and 'ASM' (Asset Management) is responsible for all operational buildings. Finally, the stakeholder structure of Schiphol will include the division ICT & ST (Information Technology & Schiphol Telematics). It is indicated that the divisions PLUS, ASM and ICT are mostly involved and important stakeholders for the implementation of BIM at Schiphol.

As can be seen in figure 15, ASM is divided in four different divisions. These divisions are; DEV (Development), P&P (Planning & Portfolio), TEC (Technical Expert Centre) and M&O (Maintenance & Operations). The division TEC is also divided in four sub-divisions, namely, TM (Technical Management), PDA (Permit, Data & Analytics), FAM (Functional Applications Management) and EP (Expert & Policy). The division PDA is divided into different teams such as the division FAM. PDA has different teams which are focusing on assets information and FAM has different divisions focusing on the different maintenance tools.

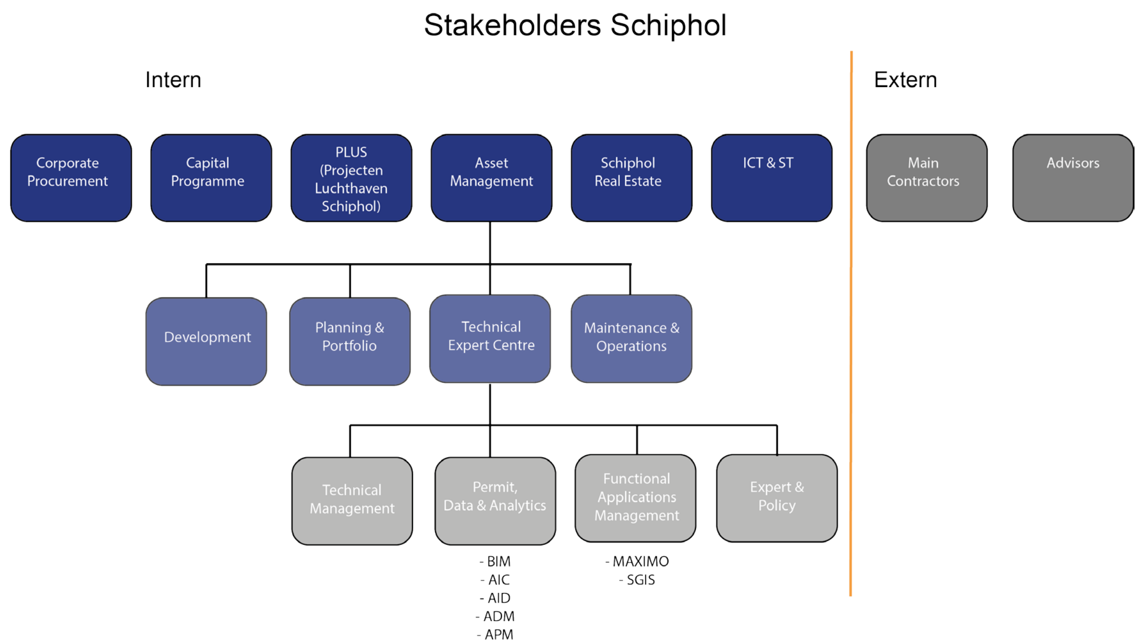


Figure 15: All stakeholders which are involved in building projects at Amsterdam Airport Schiphol.

4.3 Processes

Structured processes are an important factor, because of the diverse, big and complex projects occurring at Schiphol. When a new project will start, different divisions at Schiphol are involved. These divisions are explained in paragraph 7.2. First, the department Planning & Portfolio will draft a project description in one A4. Secondly, the department Development will take over and define the project and compose a project team. The project will be assigned to PLUS or Capital Programme, depending on the project. This division will work together with the Technical Expertise Center. TEC will provide in all information documented.

Currently, all advisors are asked to make 3D information models of the new building projects at Schiphol. Because of the relevance of BIM at Amsterdam Airport Schiphol, new processes were determined to improve the information exchange structure during projects. The new process is based on the Information Delivery Cycle of the PAS 1192-2 (Bsi, 2013) discussed in paragraph 3.3. Schiphol determined a building process with six different stages. These stages are illustrated in figure 17. In this picture the stages of Schiphol are compared to the different stages indicated by the RIBA Plan of Work 2013 (RIBA, 2013). The first stage of Schiphol is the initiative and project definition. The second stage is the design phase, where after the third phase is the tender documentation. After this decision moment the construction phase will begin, which will be concluded with a handover. Lastly, the maintenance phase of the building will start, which will actually be the longest period of the building lifecycle.

The handover of these models after the realization of the building will be an important moment. To guarantee better quality of the models during the handover, Schiphol uses the systematics of COBie data drops. A data drop is a moment to exchange information, or a 3D model. A data drop consists of a graphical model with non-graphical data attached. In addition, there is still a need for certain documentation (Bsi, 2013). Multiple moments are specified, to indicate if the models are modelled and growing correctly, because a good information model cannot be made in one time at the end of the project, see figure 16. Besides that, not all information is always needed in the model.

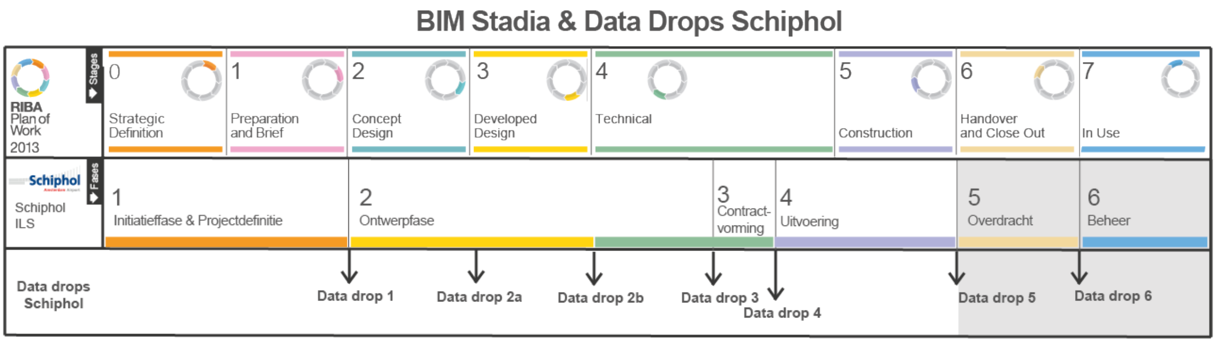


Figure 16: BIM stadia & Data drops Schiphol

To support this new process and capture all new requirements, several documents were set up by Schiphol. The different documents and the relations between the different documents are indicated in figure 17.

Schiphol has an Informatie Leverings Specificatie (ILS) (Employers Information Requirements (EIR) in English). The ILS together with the BIM protocol will define the contractual conditions and requirements which should be delivered by the contractor. This will include all the BIM extracts, drawings, documents and the ownership of these products. Also, the different processes, tasks and responsibilities are described in the ILS and will be part of the contract between Schiphol and the different suppliers. Besides these documents, the contractor should draft a BIM Execution Plan (BEP). In this plan they should capture the different agreements the contractor will make to fulfill the ILS and BIM protocol provided by Schiphol. A contract and project definition are always needed as well.

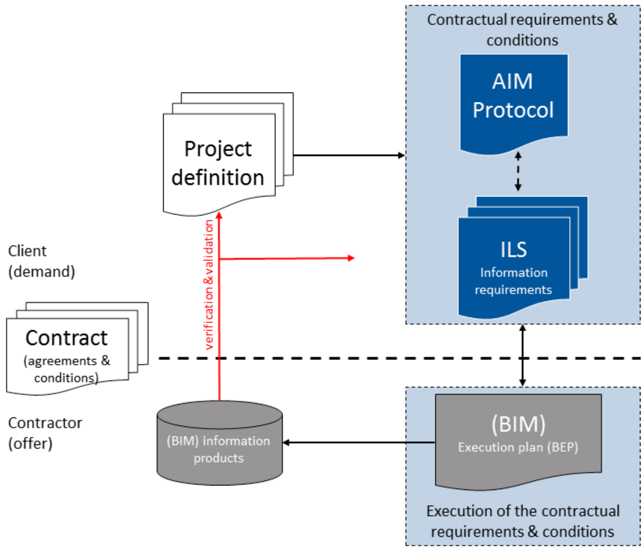


Figure 17: Different documents related to BIM at Amsterdam Airport Schiphol

When a project is finished, the handover takes place. All documents, including the BIM model, will be sent to the Asset Information Desk of Schiphol. This division will check if everything is complete. There is no check on the quality of the information. If the handover is complete, the correct documents will be sent to the divisions responsible for filling the different systems at Schiphol. All information related to the assets will be put into the facility management system MAXIMO. The information about the surroundings such as maps and floor plans will be put into the SGIS system. All remaining documents will be put into EDMS, which is a document management system. This process is captured in a BPMN diagram, see appendix I.

4.4 Interviews

To support the case study several interviews were obtained at the company. Different stakeholders at Schiphol were approached for these interviews. In this paragraph the set-up of the interviews and a summary of the results is discussed. The questionnaire of the interview is included in appendix II. By interviewing different stakeholders at Schiphol, the different opinions and needs of the departments can be reached. Furthermore, it can be made clear who is responsible for which tasks in the department structure of Schiphol. The goal of the interviews is to map the needs and requirements of Schiphol from different perspectives. Besides this, an indication is needed about how the building information set as required by these stakeholders, will be used during digital asset management. The interviews will be conducted with an open framework, which will allow two-way communication, and will provide in different opinions, but most of all provide reliable and comparable qualitative data.

The different interviews which were taken are shown in table 1. The date of the interview is indicated, as well as the name and function of the person who was interviewed.

Table 1: Overview of the interviews

Nr	Date	Person	Function
1	18-10-2017	Mohamed El-Morabit	Technical Information manager
2	23-10-2017	Sebastiaan de Sterke	Product owner MAXIMO & ICD
3	23-10-2017	Rob van Velzen & Jan Roelof Kooreman	Project manager PLUS KLM-ICA Lounge Developer ASM KLM-ICA Lounge
4	23-10-2017	Alexander Worp	Product owner/ Strategic advisor BIM
5	25-10-2017	Jos Pilon	Consultant information management BIM & SE
6	16-11-2017	Alexander Hornickel	Product owner Asset Data Management
7	30-11-2017	Kees van 't Hoog	Product Owner SGIS

4.4.1 Interview findings

During the interviews with several stakeholders at Schiphol, different opinions and needs were determined. Overall, the interviewees recognized that digital information management is needed in the future to reach the goals set by Schiphol to be and stay the preferred airport in the future.

The interviewees have different functions in the organization of Schiphol. An interview was taken with the responsible person for processing all information during the handover. Also the responsible employees for filling the different maintenance systems were interviewed, such as project managers and information managers. A distinction was made between employees who are more focused on projects and employees who are focused on maintenance and operations. These two groups differed in their needs. Mainly because people working on the projects thought more focus was given on maintenance, however they felt more effort should be put in implementing BIM during project management. People working for maintenance felt vice versa. In the future both project management and asset management should be taken into account to make sure an entirely established process will be implemented at Schiphol.

In general many of the interviewees were of the opinion that BIM can only be implemented in clear established steps after the new process for asset management is determined. This means the interviewees want to have a more clear overview of why and how they want to use the information during maintenance. If this is evident, they can better inform the contractor or advisor on why they need the information delivered according to the ILS. To reach this, more support and openness from the whole organization and management of the organization is needed. An additional need is that new processes should be ready to implement directly.

Especially, the information specification (ILS 3.0/3.1) of Schiphol is received as too complex. Many interviewees felt the information requirements were too much and not everything is used or needed. This can also be one of the reasons why the information is not complete after the handover. In the future, the specification should be aimed more on delivering the right data or information. Currently, many different documents such as 3D drawings are still asked. This handover is still focused too much on the old project processes, when the delivery was in 2D.

Besides the missing information during the handover, there is a lack of validation of the information. According to the AID they are only responsible for the completeness of the handover and not for the content. At the same time, the different divisions using the information expect that the information sent from the AID is validated and correct. This will mean in the end that there is no assurance about the quality of the information.

In the future the handover should be focused more on 3D information and on delivering the right information about objects/assets according to the requirements. Therefor an object-related database is needed to assure a better environment for maintenance and operations. An important requirement for this database will be compatibility with all assets in the future. Currently, only building assets are implementable through BIM. In the future all assets of the airport should be compatible, such as IT assets or assets related to luggage. Still some disagreement exists between using one database in the future which will fill the other source systems (MAXIMO, SGIS) or using the these source systems to fill the new database with information.

To reach the best practices not only BIM should be implemented from now onwards, but other techniques such as Systems Engineering which focusses on the requirements should be connected to BIM. By using SE practices, all requirements can be reached and this can help with the validation process of the information. An improved validation process is needed in the future. Without validation the information delivery cannot be trusted and it is difficult to indicate the responsible stakeholder. To establish a good validation process, the airport should have the right tools and intellect to do this in the future.

4.5 Conclusions inventory Schiphol

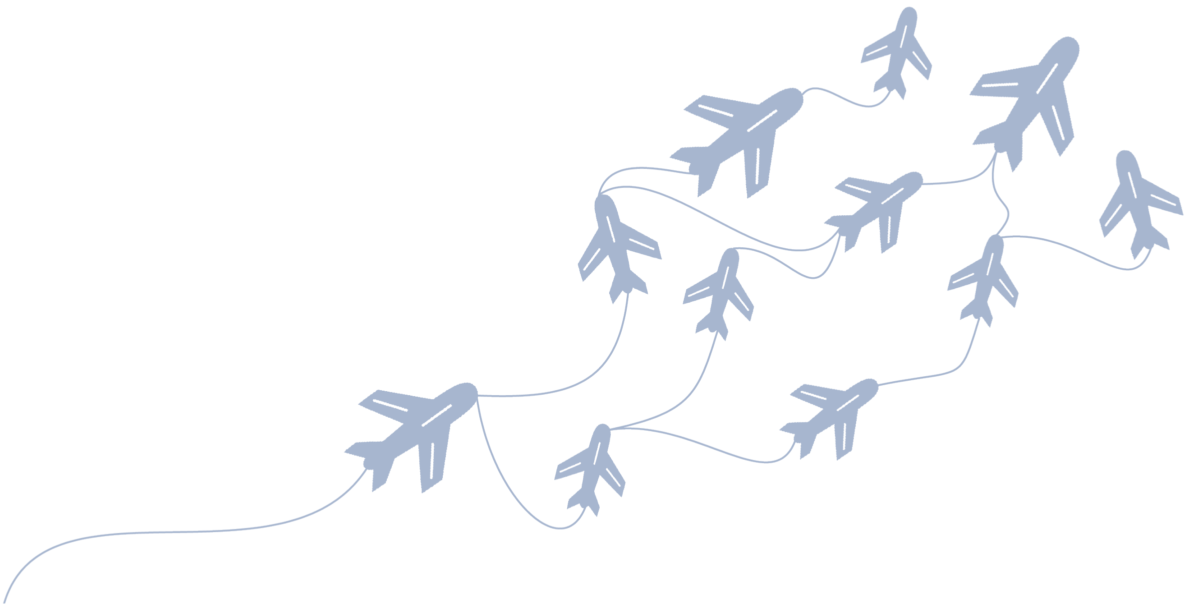
From the inventory of the processes at Schiphol different findings can be concluded. First, it can be concluded that information about assets is important to make sure daily work can be facilitated. Better information about assets can be used to reach better and more effective asset management, according to the ambition of Schiphol, explained in paragraph 4.1. Besides this, using 3D information can be used to reach a better organization structure and improve innovation and digital developments. To implement BIM and reach the ambition for the future it is important to have all stakeholders on board. The most important stakeholders are the project management, asset management and IT department. New processes should be implemented and aligned with the use of BIM. Data drops are already implemented during the processes. In the future attention should be given to structure the new processes for assimilating information during maintenance and operations.

The interviews resulted in different needs from the companies perspective about how to improve and implement BIM processes in the future. Important is to take into account that on both the project and maintenance side of the process BIM should be implemented to make sure added value is reached. A greater focus should be put on information specifics instead of deliverables in documents. Big improvements can be made by validating the information. In the future a new process or system should be established to capture the 'one single source of truth' of all assets of Schiphol. This includes building assets and other assets, such as IT assets and luggage assets as well. The new database which will serve as the 'one single source of truth' can capture data in the database or retrieve the information from other databases.

5. Methodology

This chapter describes the methodology which is used to compare the findings of the literature review and the case study. From these two chapters, the important conclusions will serve to determine the requirements for exchanging information by using BIM for asset management. These requirements will define which information exchange structure should be chosen to further elaborate on.

In chapter 5.1 the requirements for exchanging information will be determined and practical use cases will be adopted to substantiate these requirements. Besides this, the requirements will be based and complemented on the company's business requirements and on data requirements found in the literature review. In chapter 5.2 the model implementation which results from the requirements will be explained. This chapter will give insights in the new process structure and the new structure of the information in the form of an ontology. After the new structures are captured, these structures will be tested in chapter 5.3. In this chapter an application is made with the meaning to make a proof of concept. This proof of concept will establish if the chosen structures will be working. After this a validation of the proof of concept is given in the same chapter. After the tool is tested, the limitations will be explained. In the last paragraph of chapter 5, the added value for the business will be explained based the value proposition.



5.1 Determine requirements for exchanging information

During the literature research and case study different findings were stated which are interesting to further elaborate on. The literature research explains different information exchange structures. Secondly, in the case study the vision and processes of Schiphol indicate a need for change and different needs and problems arise from the interviews with the employees at Schiphol. In this chapter the current practices of exchanging data are delineated, which will also indicate the core needs of the data used. Secondly, from the business needs and problems use cases are developed, which can be put into practice to solve the problems and improve the data in the future. To substantiate the comparison of the information structures, the different needs and problems will be transferred to requirements. An information structure researched in the literature study will be further elaborated on in the next chapter. The requirements are used to define and test this new information structure.

5.1.1 Current practices of exchanging data

In the research approach is stated that the main focus of this research will be put at the handover of the building project to the asset manager. Eventually, the main purpose of using BIM at airports, is to go from descriptive data analytics to predictive data. The current situation is however not sufficient yet.

During the handover a graphical model with non-graphical data is transferred including documents. In the case of Schiphol this information is transferred to the 'AID'. This division will check if all information is indeed complete. The information, however, is not checked on quality or content during this step in the process. The AID will sent the corresponding information to the correct data platform. This documentation is divided over the 3 main systems at Schiphol, namely MAXIMO, SGIS and EDMS. The problem after dividing this information is, that no connection exists between the different systems anymore. Furthermore, the information is not validated, which means that no assurance can be given about the quality of the data. Differences between the data in the different systems occurs and no one source of truth is indicated. This will result in data which is complex, redundant, time-consuming and error-sensitive to process, however data should be complete, heterogeneous, consistent and reliable, see figure 18.

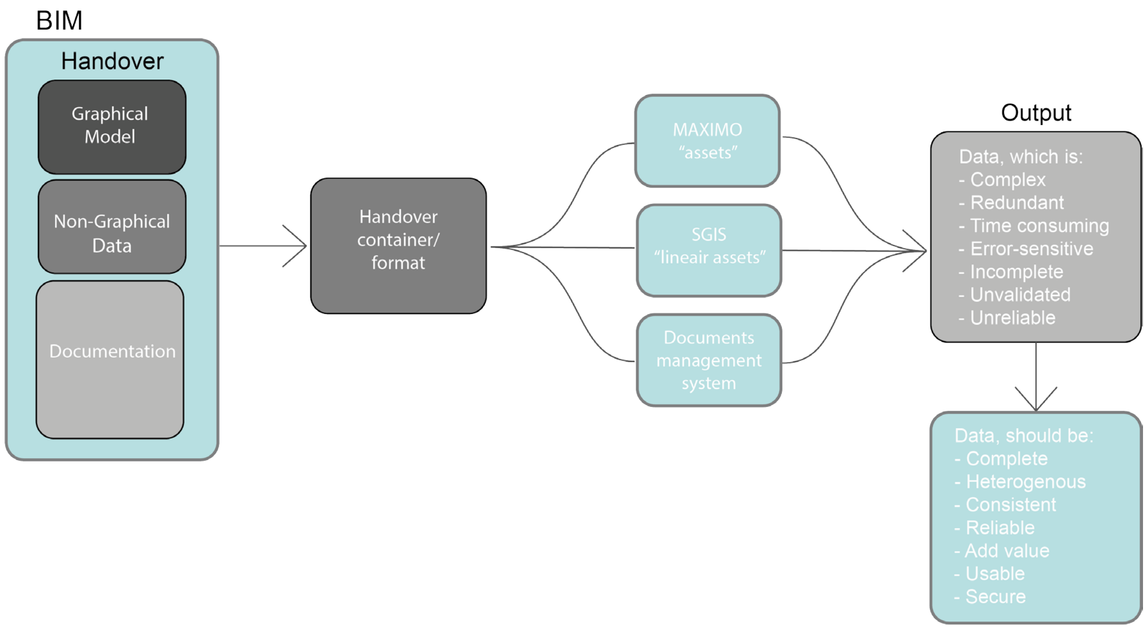


Figure 18: Handover process at Schiphol

If the data will not fulfill these requirements, no decision can be made according to this data. If the collected data can be validated, the data can be transferred into organized information. When this information will be put in the asset management systems, this information can be summarized, and knowledge exists. If the company has knowledge about their assets, the information can be analyzed, which will give insights in the information. With insights, the information can be synthesized, which will result in wisdom. With wisdom better decisions can be made and the decision making process can be optimized, see figure 19.

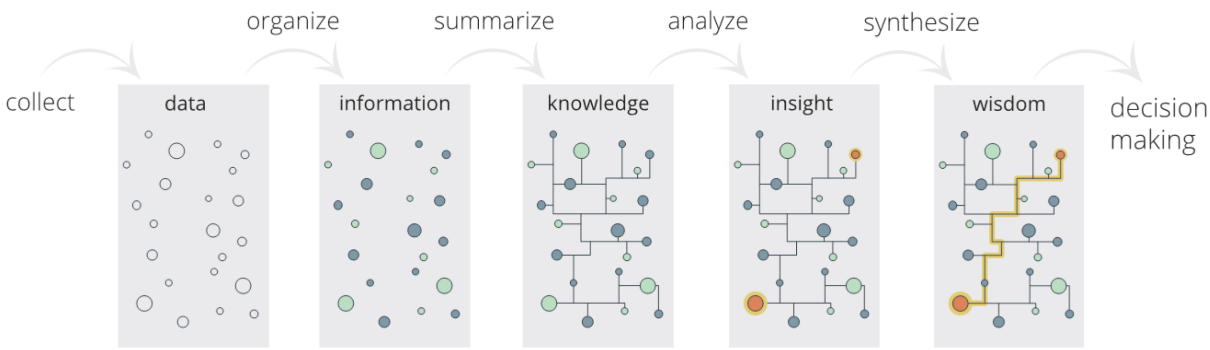


Figure 19: The purpose of collecting data to structure information, Schiphol (2018)

The data should be compliable with the following requirements as was stated in chapter 3.5, asset management. These are the **core data requirements;**

- Be complete
- Be heterogeneous
- Be consistent
- Be reliable
- Have added value
- Be usable
- Be secure

5.1.2 Business requirements

Besides the technical requirements of the data, requirements related to the business value should be taken into account. These requirements can be subtracted from the different needs of Schiphol. These needs are conducted from the interviews and can be summarized in the following points;

- Focus on both project and asset management when implementing BIM
- The new process should have concrete steps which are formulized in a clear strategy
- New techniques should be directly implementable
- Support and openness from the whole organization
- Information specification should be less complex in the future
- The information specification should be adapted. More data is needed, however less documents/products are needed in the future
- Systems Engineering should be related to BIM
- The new database should be object-related instead of document-based
- The database should be compatible with all assets, such as IT assets and luggage assets and not only building assets
- Validation and verification is needed before the information can be processed

From these needs the requirements which are important to create added value for the business in the future can be stated. These requirements support the vision of a business which would like to go from descriptive data analytics to predictive data in the future. With predictive data insights can be given into the asset information or performance. Scenarios can be made with this information. In this way, a better use of (existing) information can be created. The following business requirements are applicable;

Business Requirements;
- Clear overview of needed information per object
- Information must be checked/validated on completeness and structure
- Information must be checked/validated on quality
- Information delivery in a consistent way, which will be future-proof
- Information should be accessible
- Connection between information/data in the different systems should be available
- Implementation of new techniques or standards should be supported by a changing management process and should be clearly described
- Information should be collected and updated (automatically, when possible) over the whole life-cycle of an asset
- The asset information in the database should be related to requirements of the asset

These requirements are based on the needs of the Schiphol Airport, but can also be applied to other companies which are managing assets.

5.1.3 Asset requirements
Besides the core data requirements and the business requirements, the functional requirements of the asset should be taken into account. Every asset has different specifications and different requirements. The difference between the requirements per asset depends on the type of asset. For example, building assets differ from type from IT assets or luggage assets. This will result in a difference in needed information and is very specific to the different kind of assets. However, general requirements for assets exist, which will apply to all assets. These general asset requirements are explained below and are extracted from the literature review and case study.

General Asset Requirements;
- The asset information systems should be based on open standards
- The asset information should be easy accessible
- History or the revisions of asset information should be present
- An asset can be composed of different layers or elements. When an asset is composed of different elements, the elements should be linked
- Non-graphical data should be linked to the specific elements
- An external link to documents should be possible

Besides this, Schiphol determined generic and specific characteristics for all their assets. The generic characteristics should be available for all their assets. The specific characteristics are based on the asset and include the needed documents for this asset as well. This is a rather extended list, depending on the asset. A determination should be made if all specific characteristics should be delivered in the model during the handover.

In table 2 the generic characteristics which should be distracted from an IFC-model are indicated. In addition, some basic registration characteristics can be valuable to distract from the BIM model as well. These characteristics are indicated in table 2. The basic registration includes the GUID, which is an unique number and can only be identified from the IFC model. This is an important number, because this is the URI of the object in the BIM model. Furthermore, the name of the object and the base quantities are included in the basic registration characteristics.

Table 2: Basic registration and generic characteristics

Basic reg.	GUID
	Name
	BaseQuantities
Generic characteristics	Asset ID
	Project
	Asset description (type)
	Status
	Assettype code (STD)
	Location (Building, Storey, Space)
	Contractor/Engineer
	Supplier
	Manufacturer
	Year of construction
	Delivery date (Opleverdatum)

Use case the elevator

As an use case, the handover process of specific assets can be investigated. An example of such an asset is the elevator. In the information specifications of the assets an indication is made about all generic and specific characteristics of the assets. The elevator as an asset has the most specific characteristics. This indicates that the non-graphical data specifications at the handover will be very extensive and good to research as an example. Secondly, an elevator is built up of different elements or components, which will be forming one asset together. Furthermore, documents are important for this asset. For example, warrant specifications should be available for the elevator.

The following important questions arise for the use case about the elevator, which are in line with the sub research questions of this thesis, namely;

1. The elevator consists of different elements, which form one asset together. How should the elevator as an asset be modelled?
2. Which non-graphical data should be attached to the model of the elevator? And which documents should be linked to the model?
3. How should the handover be transferred to Schiphol?
4. Which part of the BIM (object, information and documents) will be transferred to which system? So, which data is used for what system?
5. How can the link between the data captured in the different systems be assured?

To answer these questions, a data structure should be chosen and a means to exchange the data. In this chapter a focus will be put on how the elevator should be modelled and which non-graphical data and documents should be attached to the model.

First of all, the following registration and performance questions about the elevator have to be taken into account. These questions can be related to the minimal characteristics from table 2. The relation between the questions will be explained in paragraph 5.2.1;

1. Can the elevator be identified?
2. Can the elevator be located?
3. Is the elevator up and running? (status)
4. Is the elevator maintained according to schedule?
5. Does the owner of the elevator fulfill the needed legal obligations of the country?

Because the elevator is a transport element and a system, different legally established conditions have to be taken into account during maintenance & operations. According to Stichting Stabu (2016) the following key points have to be taken into account;

- Legal obligations
- Company reliability/security
- Preventive maintenance

The following documents (in Dutch) are needed for legal obligations in the Netherlands for every elevator;

- Verklaring van overeenstemming EU richtlijn i.c.m. bouwjaar voor liften en machines gebouwd onder regime van de Europese richtlijnen
- Gebruiks-en onderhoudsinstructie
- Warenwet en Arbeidsomstandighedenbesluit includes:
    - Logboek
    - Merk van goedkeuring
    - Certificaat van goedkeuring van NL-CBI voor liften
    - Bewijs van goedkeuring door “deskundige” voor machines/arbeidsmiddelen
    - Keuringsrapport

Besides this the following technical characteristics of the elevator should be included;

- Type of elevator (passenger lift yes/no)
- Propulsion (Aandrijving NL)
- Capacity by weight/number
- Velocity
- Amount of stopping places
- Lifting height

5.1.4 Requirements compared

In the previous paragraphs the different requirements related to the data (§5.1.1), the business (§5.1.2) and the asset (§5.1.3) were stated. These requirements are gathered in figure 20. The core data requirements are applicable to all information about an asset and to every other requirement. For example, the first business requirement recommends that the information should be presented in a clear overview per object. This overview should of course be complete, consistent, reliable etcetera. The business requirements, however, are more generic and the asset requirements are more specific related to an object. The business requirements should make sure the information in the management systems should improve the working processes and structure the information processes and the asset requirements should make sure the information for every asset is present in the management systems.

It is important to take all the different requirements in account, because of the changes in the processes during the implementation of BIM. The requirements are based on construction industry requirements, asset management requirements and IT requirements. The construction, asset management and IT sector will all be important for the implementation of BIM. Data will play a bigger role in the future and it is recommended to use an open standard, such as linked open data to implement. By using a 'small' use case, the technique to implement these requirements can be made clear. Such a use case can be rehearsed to implement the techniques on a bigger scale. In this case a specific asset, the elevator, will be chosen as the use case. After proving successful the use case can be repeated and implemented in the same way for other assets.

Business requirements

- Clear overview of needed information per object
- Information must be checked/validated on completeness and structure
- Information must be checked/validated on quality
- Information delivery in consistent way -> future-proof
- Information should be accessible
- Connection between information/data in the different systems should be available
- Implementation of new techniques or standards should be supported by a changing management process
- Information should be collected and updated automatically over the whole life-cycle of an asset
- The asset information in the database should be related to requirements of the assets

Core data requirements

- Completeness
- Heterogeneous
- Consistency
- Reliability
- Added value
- Usability
- Security

Asset requirements

- An asset can be composed of different layers or elements. As an asset is composed of different elements, the elements should be linked
- Non-graphical data should be linked to the specific elements
- An external link to documents should be possible
- The asset information systems should be based on open standards
- The asset information should be easy accessible
- History or the revisions of asset information should be present

Figure 20: Business, data and asset requirements

5.2 Model implementation

In chapter 4 the different needs of Schiphol are explained. These needs were transformed in requirements in chapter 5.1. The result of this thesis will provide a new information structure which will implement the stated requirements. In this chapter the model implementation which results from the requirements will be explained. The model implementation will provide in a new process structure. Finally, this chapter will give insights in the new structure of the information in the form of an ontology. The ontology will be explained and different extension options for the ontology will be given.

According to current processes of Schiphol, the information at the handover is stated in the ILS of Schiphol. This ILS asks for many different files besides the BIM. Examples of such files are 2D drawings and documents for legal obligations. The graphical and non-graphical information are mostly captured in the IFC file. Documents should be delivered in separate file formats such as PDF and Excel. When these files together are transferred to the company a container format should be used, to combine the information. According to the literature study, the ICDD container is currently the best information exchange structure available. The ICDD container can be implemented to link the documents based on the objects present in these documents with the use of semantics. This will result in a format which is based on linked open data.

To use the semantics of this handover container, a new environment should be set up based on linked open data. This new environment which will be a CDE should be object-based and a clear information structure should be fundamental to the environment. This information structure can be a User Defined Ontology (UDO) which captures the requested data by Schiphol in the data hub and the connections between the data. The Schiphol Data Hub can be compared to a CDE. From this CDE, the current or new management systems can be filled by selecting the parts of the information needed for this system. This process structure is sketched in figure 21.

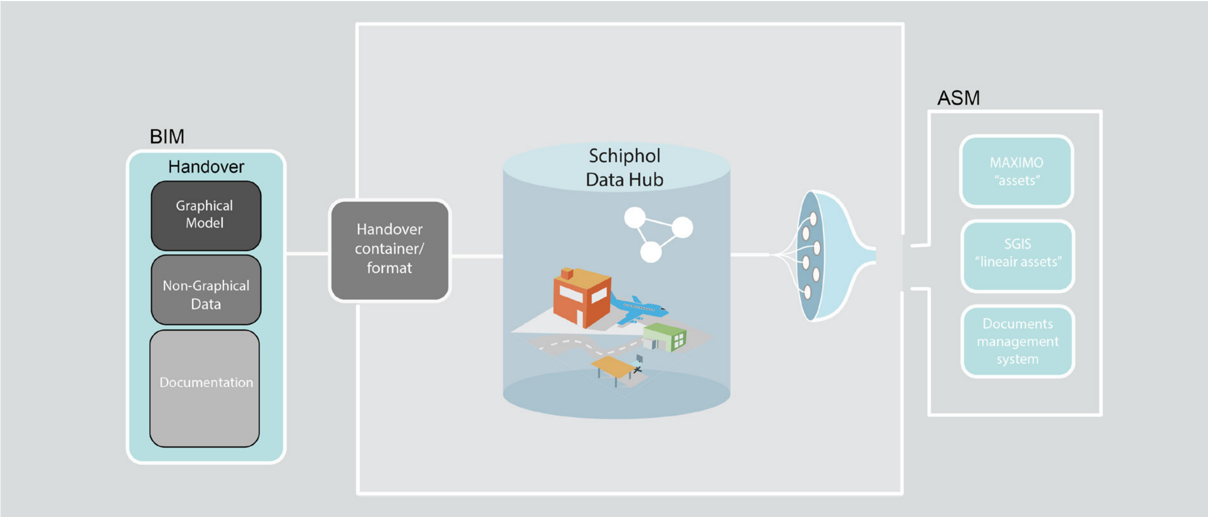


Figure 21: New process structure

This research focuses on the transfer, validation and use of the information for asset management. Important parts of the model are the handover container, the CDE with the according information structure and the validation process. According to the literature research (paragraph 3.4) and requirements (paragraph 5.1) the information exchange structure and CDE can both be based on linked open data. If these systems are both based on linked data, exchanging information between these two formats will be easier and less complicated.

The first step is to define and structure the CDE. Without a structured CDE, the handover container cannot be implemented in a system afterwards. No software implementation for this CDE currently exists. To start implementing the first step towards the data hub, the information structure should be determined. This information structure should be based on an ontology, which can structure the data according to the requirements set in the previous chapter (paragraph 5.1). To implement the information at the handover in the data hub, this data should be transformed to linked open data in the RDF format.

5.2.1 User Defined Ontology

Before implementing a CDE to organize all information for asset management, an information structure which determines the links between all the data and captures the information, should be constructed. An ontology can define such an information structure. Because many current ontologies exist, these will serve as a starting point to construct an UDO. An UDO is sometimes also referred to as an Object Type Library (OTL).

Currently, an ontology structure is under development at Schiphol. The basic structure of this ontology is based on system engineering requirements. This structure can be seen in figure 22 and is constructed with the software program Protégé. The RDF file of the ontology can be found in Appendix III. The ontology created for Schiphol, uses the prefix "aas" which refers to the name of the ontology, 'Amsterdam Airport Schiphol' ontology. The ontology has the following Unique Resource Identifier (URI); **prefix aas: http://www.schiphol.nl/ontology#**

The ontology consists of the following main classes; aspect, location, requirement, information item, characteristic and object. The main class object includes all objects, such as building elements and transport elements. This class can be mapped with the ifcowl class IfcElement. The main class aspect includes the compositions of one or more objects, such as assets, components and systems. The class location stores the site, building, storey, space and zone where an object is located. The class characteristic and requirement store properties and attributes of objects. Characteristics can be the GUID, name or other properties of an object. The class requirement contains requirements which are needed for example for legal obligations, such as warranty information. The last main class Information item will store functional requirements, such as a status of the asset. All classes have many subclasses, but by using more main classes the structure becomes not very hierarchical. This gives more opportunities for querying.

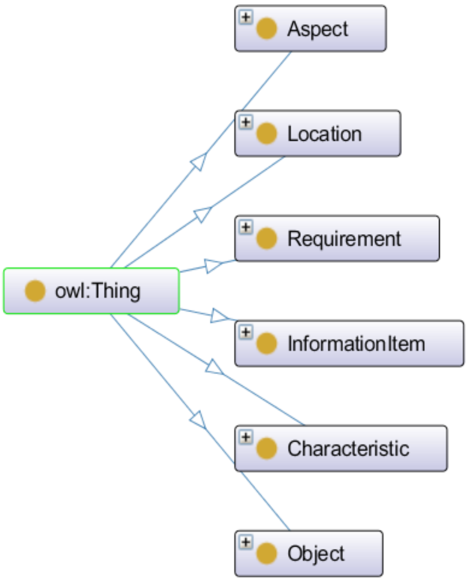


Figure 22: Basic structure ontology of Schiphol (2017)

After identifying the main classes, the ontology can be explained when extended with assets. The first asset added is the elevator, which is proposed as an use case in paragraph 5.1.3. In this paragraph some minimal identification and performance questions were given about the elevator, which can be used to state which information is minimally needed about an asset for maintenance & operations. These questions were;

1. Can the elevator be identified?
2. Can the elevator be located?
3. Is the elevator up and running? (status)
4. Is the elevator maintained according to schedule?
5. Does the owner of the elevator fulfill the needed legal obligations of the country?

Almost all these questions are answered after checking the basic registration and the generic characteristics for all the assets based on ILS of Schiphol. The comparison between the questions and the characteristics are made in table 3. In the last column of the table the questions with the numbers are indicated.

Table 3: Minimal characteristics required for an asset compared to the asset requirements

Basic reg.	GUID	Identify (1)
	Name	Identify (1)
	BaseQuantities	Identify (1)
Generic characteristics	Asset ID	Identify (1)
	Project	Identify (1)
	Asset description (type)	Identify (1)
	Assettype code (STD)	Identify (1)
	Status	Status (3)
	Location (Building, Storey, Space)	Location (2)
	Contractor/Engineer	Maintenance (4)
	Supplier	Maintenance (4)
	Manufacturer	Maintenance (4)
	Year of construction	Maintenance (4)
	Delivery date (Opleverdatum)	Maintenance (4)

- **Adding identification characteristics to the ontology**

First, the identification characteristics of the elevator are implemented in the ontology. Part of the schema of the ontology of the elevator is illustrated in figure 23.

The elevator is a subclass of the type IfcTransportElement, which is a subclass of the main class object. The class object can be aligned with the class IfcElement stated in the IfcOWL ontology. To identify the asset, the elevator refers to the class characteristics. Subclasses of characteristics are; GUID, measurement, name, type and STD. The second objective was to locate the asset, which can be done with the class location. The elevator refers to this class as well. A subclass of the location is the class zone, which is aligned from the BOT ontology. This ontology also includes the building, storey, space and site.

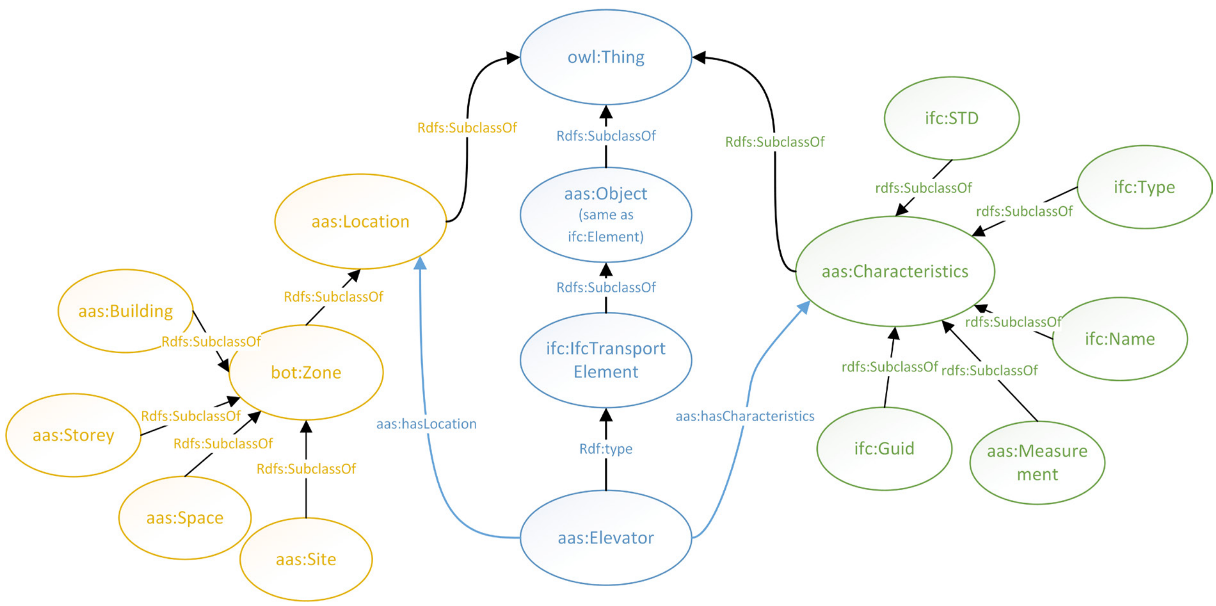


Figure 23: Part of the ontology of Schiphol based on the elevator

- **Adding the status of an asset to the ontology**

Besides adding characteristics and the location to the asset, the status of the asset can be of importance. Because all assets, and not only the elevator, should have a status, the main class object should refer to the class status. If a main class refers to a certain value, this means that all subclasses will have that value as well. The relation between the objects and the status is illustrated in figure 24. The class status has different subclasses, which indicate the current status of the assets. Examples of such statuses are; in use, new, existing, temporary and demolished.

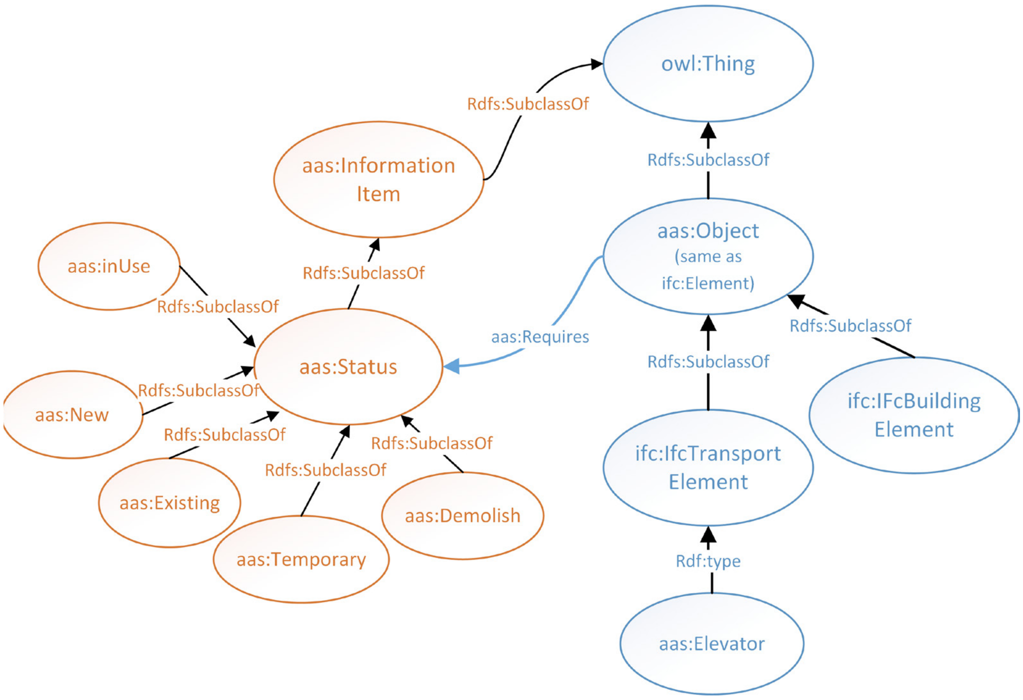


Figure 24: Extension of the ontology with a status

- **Identifying classes as an asset and adding subclasses**

Because the asset elevator normally exists of multiple parts which will form together the asset elevator. This means that the asset is a collection of multiple objects. To model this in the ontology, different subclasses of the elevator will be initiated, such as the elevator cabin, door, shaft and counterweight. The class object indicates that an object can contain another object. So, the elevator can contain the subclasses of the elevator. The elevator is linked to the class asset, which is a subclass of aspect. With this link the filtering can be made between assets and the objects which belong to a collection of objects. The superclass which will store the collection of assets, should be the class related to the class asset. This part of the ontology is illustrated in figure 25.

Information about maintenance is also needed for asset management. This part includes information about the contractor or engineer, supplier, manufacturer, year of construction and delivery date (opleverdatum). The ontology can be easily extended by adding this information as subclasses to the main class 'characteristics'. This can be constructed equal to the adding the identification characteristics.

The last important requirement to fulfill in the ontology is that documents should be linked to the object. To reach this, a class can be made in the ontology which states the kind of documents which should be linked. When the information is added to the ontology, a web link can be

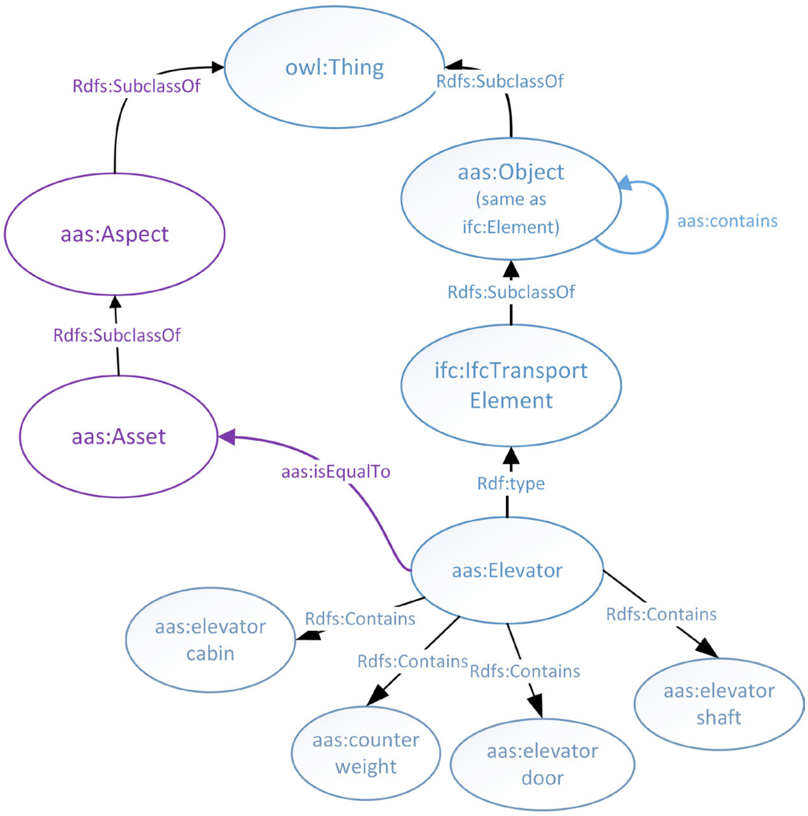


Figure 25: Extension of the ontology with a collection of objects which form one asset

saved in this class instead of the information. A second option is to store the information in the UDO with the help of semantics. This can be done with the help of the ICDD, which provides a solution to link documents and information in documents to objects in a model with the help of linked open data. To complement this in the UDO more additional research into this topic should be done.

5.2.2 Extending the ontology

Besides the elevator, other assets can be added to the ontology, which makes this technique more flexible than other techniques. This research focusses on assets which can be retrieved from BIM models, but in the future other assets which are assimilated from other sources should be added. This is an important factor at Amsterdam Airport Schiphol, because during asset management all assets and the information about these assets should be accessible in the management systems. Extending the ontology can be modelled in a general manner. To illustrate this, two examples will be given. The extension can be produced for all objects which can be identified from a BIM model. BuildingSMART recognizes the following types of elements (BuildingSmart, 2018);

- IfcBuildingElement
- IfcCivilElement
- IfcDistributionElement
- IfcElementAssembly
- IfcElementComponent
- IfcFeatureElement
- IfcFurnishingElement
- IfcGeographicElement
- IfcTransportElement
- IfcVirtualElement

In the example of the elevator, focus was put on the type IfcTransportElement. In the following examples a focus will be put on the type IfcBuildingElement.

- **IfcDoor**

The first asset which will be assimilated in the ontology will be the door. A door is a building element and will be included in every building model. The door is a very standard element, but has some additional properties compared to the elevator. These properties have to be added to the ontology. The process of extending the ontology with the new asset is shown in figure 26. The door is added as a type of ifcBuildingElement. In the class characteristics, new properties are added, which are linked to the door. Because not all subclasses of characteristics are applicable to all building elements or transport elements, the properties should be linked individually to the class ifcdoor instead of to the main class characteristics. This should also be done to all other elements, such as the elevator.

(Not all properties and relations which were applicable to the elevator are present in figure 26, but can also be applicable to the door. These relations are omitted to make the figure more clear.)

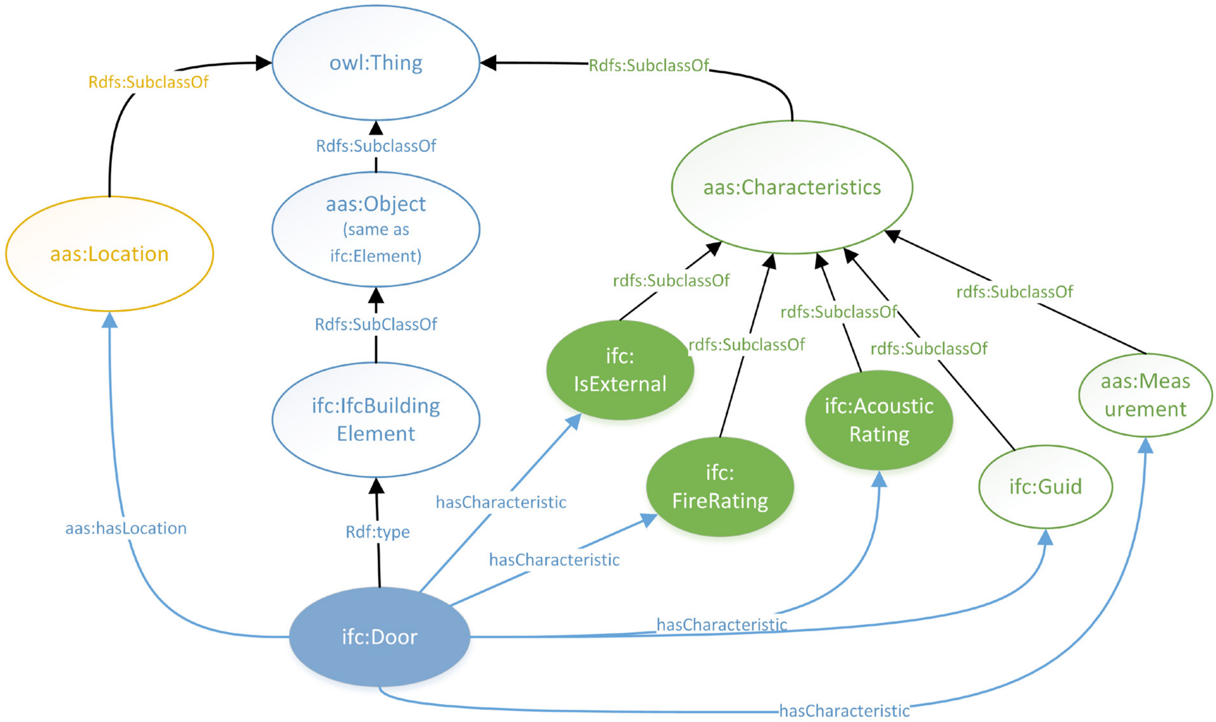


Figure 26: Extension of the ontology with the asset door

- **IfcWall/IfcCurtainWall**

The second asset which will be assimilated in the ontology will be the wall and a variance on this asset, the curtainwall. This asset has some important characteristics and a distinction between the two assets can be made. The wall has namely the characteristic loadbearing, however a curtainwall can not be loadbearing. The extension of the ontology is shown in figure 27. This example shows that these objects can be added to the ontology similarly as the elevator and door. An addition has been made to this example, because a door can be related to a wall. A wall or a curtainwall can namely contain a door. This relation can be found in IFC and can be added to the ontology as well.

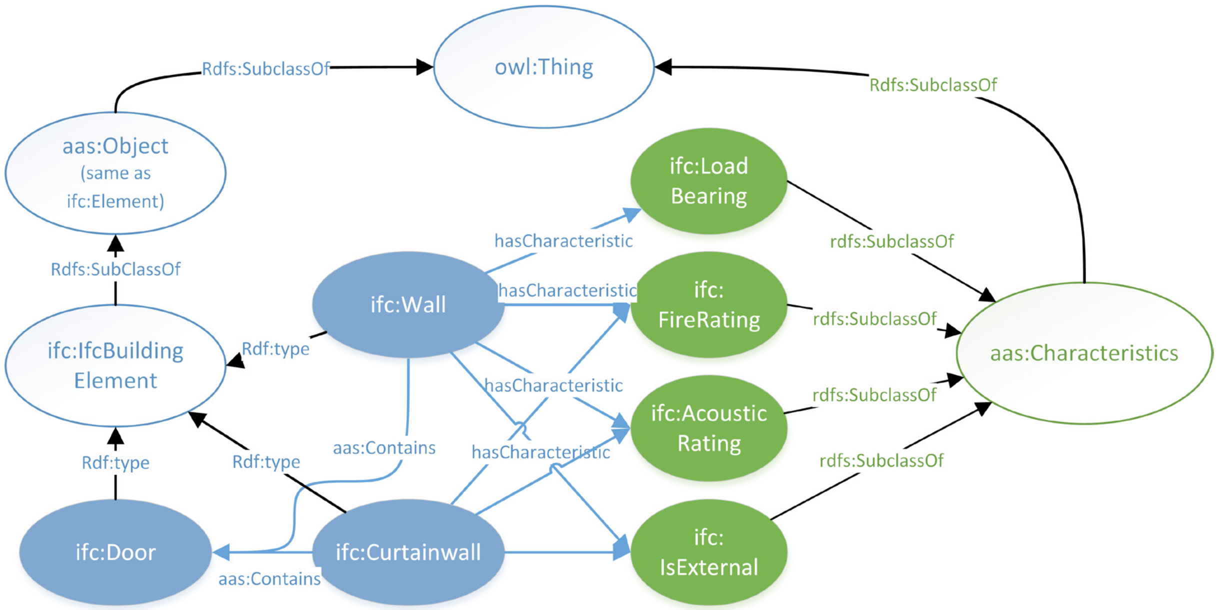


Figure 27: Extension of the ontology with the assets wall and curtainwall

5.3 Proof of concept

After constructing the UDO, the information structure of the linked open data hub is determined. The next step is to test the ontology and implement the process steps from the handover. This means that the BIM models and information which is transferred, should be implemented in the data hub/CDE. To make this transfer possible, the data from the BIM models should be constructed to linked open data. The literature study showed the potential of using linked open data for future practices. The goal of the proof of concept is to demonstrate the usage of the ontology and give practical examples of transferring and validating the as-built information. To make this demonstration an application is built, which is described in section 5.3.1. In this section the main goals of the tool, the different steps taken to reach these goals and how the application is working are explained. Secondly, a more extensive elaboration is given about the RDF data retrieved from the application and how to query this data. In paragraph 5.3.2 a validation of the proof of concept is given which will indicate how well the tool works and the limitations of the tool are given. Finally, the findings and conclusions of the proof of concept are summarized in paragraph 5.3.3.

5.3.1 The application

To goal of the proof of concept is to demonstrate the usability of the ontology and show how the information from an IFC model can be converted to a linked open data platform. Linked data principles are used to convert the information of a BIM model to a RDF format. The UDO will serve as the underlying structure of the information. According to this structure the data will be stored and validated. The validation will give an output which will tell the asset manager if the information in the BIM model is according to the requirements of the company. The information should be correctly in the file, otherwise the data will not be transferred. Secondly, the information will also be validated on correctness according to the requirements set to the data. This will show the workability of linked open data in the format RDF for 3D asset management in the future.

For the development of the proof of concept an application is composed. Parts of the Python source code from (Fernal Berrer, 2017; Stancheva, 2017) were used as a basis for the development of the application. To make this application the following programming software was used: Python 2.7, in combination with the following libraries;

- PyQt 4
- IfcOpenShell 2.7
- RDFLib 4.2.2
- OpenPyXL 2.5.0

To retrieve the correct results, several steps are taken in the application. These steps are shown in the process map of the tool, which is illustrated in figure 28. The total python script of the tool can be found in appendix V.

When the application is started, the Graphical User Interface (GUI) of the tool is shown. The images of the GUI's can be seen in appendix IV. The GUI will show two tabs. The first tab of the GUI will show three buttons. In this tab an IFC model can be loaded with the first button and the RDF file, which will be the UDO, can be loaded in the application with the second button. The third button can be pressed after these files are both uploaded in the application and starts the tool. The second tab of the GUI will give some explanation about the application. This explanation can be read when using the application for the first time.

After the application has been started, it will first read the IFC model with IfcOpenShell and all relevant IFC elements will be saved in a list. This will improve the speed of the application, because otherwise the application has to iterate over all elements after every next step. Relevant elements to save in the list will be transport elements and building elements according to the indicated classes in the ontology. The extraction of the IFC elements will be more elaborately explained later in this chapter.

The second step is to parse the RDF file which includes the UDO in the tool with the help of the library RDFlib. Parsing the RDF file will create a graph with the information structure of the ontology in the application. After parsing the RDF file, the IFC instances which are saved in the list will be linked to the ontology. If an IFC instance is specified in the ontology, the information about these instances will be added to the graph. This means that a new RDF file will be created where as-built information will be stored in RDF triples. This step will create the first output file of the application, namely the RDF file.

After the RDF file is created, this file will be used for the following steps. This part of the application will make use of the linked open data structure and the RDF file which is saved in the file format turtle syntax (.ttl). In the next step a query will run over the RDF data and will retrieve all elements with their properties and relations. The results of this query or multiple queries will be saved in a CSV file. This will be the second output of the application. An elaboration on the queries and the query results will be given later in this chapter.

To generate a better readable result from the application, the last step will be to generate a validation report. This report will use the query results and will check if the values are correct. If a value is not filled, the result will be marked red. When a value does not have the correct datatype (Boolean, string, number) the value will be marked orange. This will give a light validation checker built in the tool. The final result of the application will be an excel file, with the results and a lay-out.

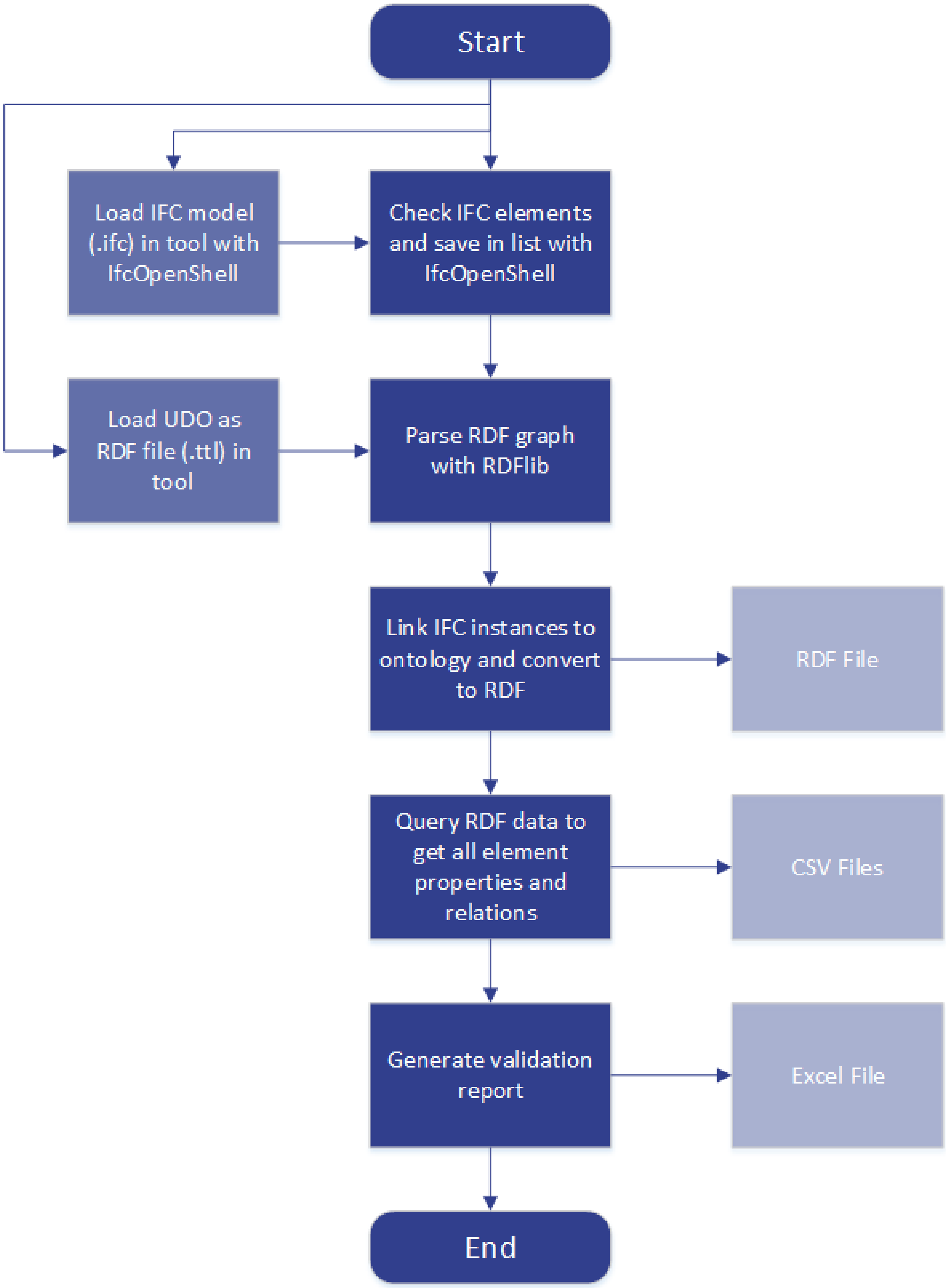


Figure 28: Process map of the application

- **Finding the information with IfcOpenShell**

The first step of the application is opening the IFC model in the tool. This can be the IFC model of your own choice. After opening the model with the Python library IfcOpenShell, the application will select all relevant IFC elements which are indicated. After saving these elements in a list, the needed properties and relations are identified. The properties and relations of the elements are indicated according to the information in the IFC file or the IFC reference site according to the IFC schema. For example, the relations between elements (IfcProduct) and spaces, building storeys and the building are illustrated in figure 29.

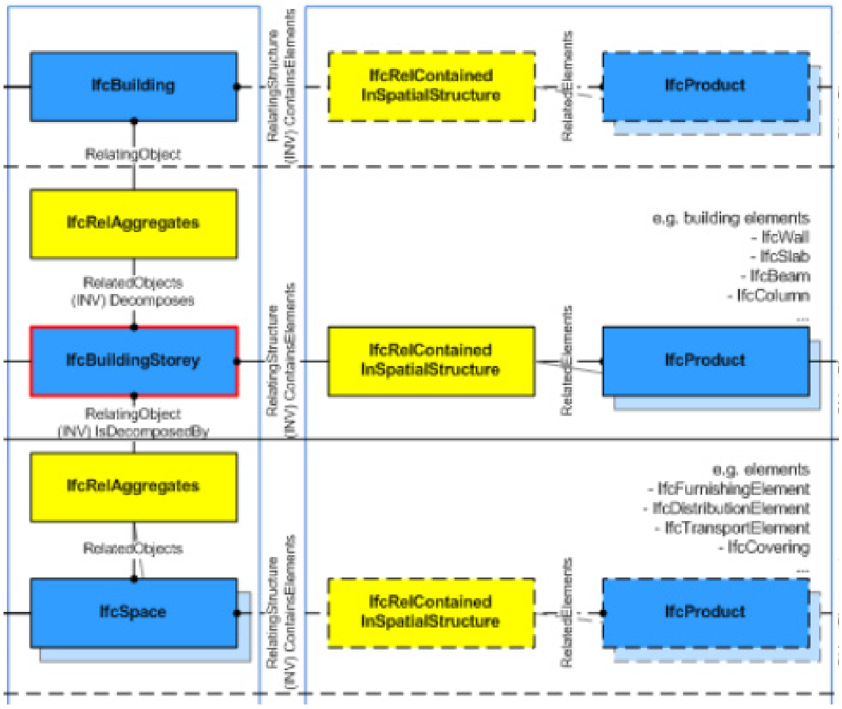


Figure 29: Spatial structure of IfcProduct, IfcSpace, IfcBuilding related to the IfcBuildingStorey in the IFC schema

Besides finding the relations between the elements, the attributes and properties of every element are searched for as well. There exists a difference between finding the attributes and the properties of a product. The attributes are always defined for every element, these are the standard identification units of an entity. The attributes are defined as in the following line of the IFC file;

#6652=IFCDOOR('1hOSvn6df7F8_7GcBWlR', #33, 'M_Single-Flush:1250mm x 2010mm:1250mm x 2010mm:146596', $,'1250mm x 2010mm', #6651, #6646,'146596', 2.0099999999999999,1.25);

In this example an IfcDoor is given. In this line all attributes are defined between the brackets. The first attribute is for example the GUID and the second attribute is a reference to another line in the IFC file, namely line #33. If an attribute is replaced by a '$', this attribute is not indicated in the IFC file. When searching for these attributes in the file, these attributes will be found, but the value of the attribute will give an empty field.

Secondly, properties of the elements are indicated in the file. These properties can be found by finding relating property sets of the elements. The property sets will refer to an ifcpropertysinglevalue, which indicates the specific property and its value. This is shown in the following lines as an example;

#6551=IFCPROPERTYSET('1DXp8U0yT8Qu9wBZ7UEYIP',#33,'PSet_Revit_Identity Data',$,(#6539));

#6539=IFCPROPERTYSINGLEVALUE('Mark',$,IFCLABEL('10'),$);

- **Querying the information**

After finding all entities, attributes and properties, this information will be saved in a graph in the RDF file. One of the following steps in the application is running a query. A query will retrieve data from an information database which includes information in the RDF format. The query language used in the application is SPARQL. The query will be selecting information from a graph.

In the script no information database is used, but the information is saved locally in an RDF file. This RDF file contains a graph with triples. The query runs over the RDF data in the turtle syntax file and will retrieve all elements with their properties and relations. The results of this query or multiple queries will be saved in a CSV file. This is an important part of the application, because by using queries more extended searches can be done. Besides this, using and exchanging information can be executed in a more efficient way which will provide in a better information reusability. The following examples show the results of different queries which can be runned using the RDF file of the application. Example 1 shows a query which will retrieve the basic information asked to identify an object. The identification requirements were explained in chapter 5.2.1. The query runs over all elements (called bEl) and returns with the related properties which include the global identifier, name of the object, object type, STD code and project name.

The query of example 1 will retrieve the results shown in table 4. Because of the extensive list of results, two elements are selected and identified in table 4. One object of the results is a door and one of the objects is a transport element. The objects which are found by the query both have a number which is retrieved from the URI. The script adds an unique number to every object, to identify every element individually. These numbers can be adapted to the specific requirements of the unique object ID's of the asset manager.

Secondly, the column of the STD code is empty, because this value was not found by the query. This means that the IFC model doesn't include an STD code or this code has been put in the wrong place in the model. The project name is for both objects the same. This should be the same for all values in the IFC model, because only one project is included in the model.

Example 1:

```
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

DISTINCT ?bEl ?GlobalId ?Name ?ObjectType ?STD ?Project
WHERE {
        ?bEl aas:hasGuid ?GlobalId .
        ?bEl aas:hasName ?Name .
        ?bEl aas:hasType ?ObjectType .
        ?bEl aas:hasSTD ?STD .
        ?bEl aas:project ?Project .
} ORDER BY ?bEl
```

Table 4: Part of the results of the query in example 1

OBJECT ID	GUID	NAME	OBJECTTYPE	STD	Project
#IfcDoor001	0CQpf9SY1DgOU1hBpiJV	66_KON_MOD_Rechte muurkap:66_KON_MOD_Muurkap (Zincorplaat):401224	66_KON_MOD_Muurkap (Zincorplaat)		Schiphol parkeergarage P3
#IfcTransportElement015	0CQpf9SY1DgOU1hBpiIM	66_KON_MOD_42306675-lift 2:66_KON_MOD_42306675-lift 2:401153	66_KON_MOD_42306675-lift 2		Schiphol parkeergarage P3

Example 2 shows a query which will retrieve the basic information asked to locate an object. The query runs over all elements (called bEl) and returns with the related properties which include the building, the specific building storey and the space where the object is located in.

The results of the second example are shown in table 5. Because of the extensive list of results, the same two elements are selected and identified as in the first example. The query retrieves the building the objects are located in, which is a parking garage and the building storey of the objects which are both on the ground floor. No spaces are modelled in this IFC file, so no spaces are retrieved by the query.

Example 2:

```
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?bEl ?Building ?Storey ?Space
WHERE {
        ?bEl aas:hasBuilding ?Building.
        ?bEl aas:hasLocation ?Storey .
        ?bEl aas:inSpace ?Space .
} ORDER BY ?bEl
```

Table 5: Results of the query in example 2

OBJECT ID	BUILDING	BUILDINGSTOREY	SPACE
#IfcDoor001	Parkeergarage P3	00 begane grond	
#IfcTransportElement015	Parkeergarage P3	00 begane grond	

The third example shows a query where additional properties are added. These properties can give information about whether the object is external, load bearing or which fire or acoustic rating the object has. When defining the ontology, it was stated that not all objects will have these properties and some objects have only a few of these properties. To make a working query, all these additional properties are added as an optional selecting of the property. When this is marked as optional, the object does not need to have these properties and all objects which only have a few of these properties will be added to the query instead of excluded based on not fulfilling all requirements. The fourth query can be read beneath and the results of this query can be found in table 6.

Example 3:

```
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?bEl ?GlobalId ?IsExternal ?LoadBearing ?FireRating ?AcousticRating
WHERE {
    ?bEl aas:hasGuid ?GlobalId .
    OPTIONAL {?bEl aas:isExternal ?IsExternal . }
    OPTIONAL { ?bEl aas:loadBearing ?LoadBearing . }
    OPTIONAL {?bEl aas:fireRating ?FireRating . }
    OPTIONAL {?bEl aas:acousticRating ?AcousticRating . }
} ORDER BY ?bEl
```

Table 6: Results of the query of example 3

OBJECT ID	GUID	ISEXTERNAL	LOADBEARING	FIRERATING	ACOUSTICRATING
#IfcDoor001	0CQpf9SY1DgOU1hBpiJV	false			
#IfcTransport Element015	0CQpf9SY1DgOU1hBpilM				

The fourth example show the addition of quantities to the query. The quantities added are the width, height and depth. The results of this query are shown in table 6. Clear width and clear height are standard properties of the property set doorCommon, which gives a high likeability that these values are correctly indicated in the model. The transport element does not have this property set, which means the values can be found in an 'ifcpropertysinglevalue'. These values are not always filled in the model and can be wrong or unclear.

Example 4

```
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?bEl ?GlobalId ?ClearWidth ?ClearHeight ?ClearDepth
WHERE {
    ?bEl aas:hasGuid ?GlobalId .
    OPTIONAL {?bEl aas:ClearWidth ?ClearWidth . }
    OPTIONAL { ?bEl aas:ClearHeight ?ClearHeight . }
    OPTIONAL {?bEl aas:ClearDepth ?ClearDepth . }
} ORDER BY ?bEl
```

Table 7: Results of the query of example 4

OBJECT ID	GUID	CLEARWIDTH	CLEARHEIGHT	CLEARDEPTH
#IfcDoor001	0CQpf9SY1DgOU1hBpiJV	1320.0	2410.0	
#IfcTransport Element024	0CQpf9SY1DgOU1hBpilS		2050.0	

The total query used in the application is given in appendix VI.

Besides running the queries to find the generic characteristics, a more in depth search can be generated for an asset. With the help of this search specific characteristics will be taken into account as well. The query in example 5 will show more results and a distinction can be made between the transport elements which actually include the asset elevator and the transport elements which include parts of the elevator. In table 8 the results of this query are shown. In this result the one object which can be identified as one of the two elevators in the model is given.

Example 5

```
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?bEl ?GlobalId ?Status ?Manufacturer ?DeliveryDate ?PassengersElev ?Propulsion ?CapacityNr ?CapacityWght ?Velocity ?StopPlaces ?LiftHeight
WHERE {
    ?bEl aas:hasGuid ?GlobalId .
    OPTIONAL {?bEl aas:status ?Status . }
    OPTIONAL {?bEl aas:manufacturer ?Manufacturer . }
    OPTIONAL {?bEl aas:opleverdatum ?DeliveryDate . }
    OPTIONAL {?bEl aas:passagierslift ?PassengersElev . }
    OPTIONAL {?bEl aas:aandrijving ?Propulsion . }
    OPTIONAL {?bEl aas:capacitynr ?CapacityNr . }
    OPTIONAL {?bEl aas:capacitywght ?CapacityWght . }
    OPTIONAL {?bEl aas:snelheid ?Velocity . }
    OPTIONAL {?bEl aas:stopplaatsen ?StopPlaces . }
    OPTIONAL {?bEl aas:hefhoogte ?LiftHeight . }
    } ORDER BY ?bEl
```

Table 8: Results of the query of example 5

OBJECT ID	GUID	STATUS	MANUFACTURER	DELIVERY DATE	PASSENGERS ELEVATOR	PROPULSION
#IfcTransport Element015	0CQpf9SY1DgOU1hBpilM	New Construction	KONE		True	KDM90
OBJECT ID	**GUID**	**CAPACITY BY NR**	**CAPACITY BY WGHT**	**VELOCITY**	**NR OF STOP PLACES**	**LIFTING HEIGHT**
#IfcTransport Element015	0CQpf9SY1DgOU1hBpilM	24.0	0.0	2.0	6.0	17690.0

- **Validating the information**

After running the queries, the results are shown in a CSV file. The last step of the application will be to validate the information and generate a validation report. This report should present a better readable result. The report will use the results of the query and will check if the values are correct. The final result of the application will be an Excel file, which shows the results in a table and will provide a lay-out for the report. When no values are marked as wrong, the results can be used for the asset management system.

The previous paragraph already indicated that some values are shown empty, because this information cannot be retrieved from the IFC file. If a cell in the table will be empty or will only show '[]' or ' "" ' the value is not filled. This will result in an error and the cell will be marked red. A red marking will mean that there should be a value in the cell which should be updated. This will result in a report which looks like the example illustrated in figure 30. For this example, the same model and queries are used as in example 1 and 2 of paragraph 5.3.1. The results are shown in a table and the empty cells are shown in red. This is applicable to the properties STD and space in this example. These properties were not included in the IFC model.

Validation Report Name Model

2018-02-20, 20:26

OBJECT	GLOBALID	NAME	OBJECTTYPE	STD	PROJECT	BUILDING	STOREY	SPACE
http://www.schiphol.nl/ontology#IfcDoor001	0CQpf9SY1DgOU1hBpiJV	66_KON_MOD_Rechte muurkap	66_KON_MOD_Muurkap (Zincorplaat)		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcDoor002	0CQpf9SY1DgOU1hBpiJU	66_KON_MOD_Rechte muurkap	66_KON_MOD_Muurkap (Zincorplaat)		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcDoor003	0CQpf9SY1DgOU1hBpiJT	66_KON_MOD_Rechte muurkap	66_KON_MOD_Muurkap (Zincorplaat)		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcDoor004	0CQpf9SY1DgOU1hBpiJS	66_KON_MOD_Rechte muurkap	66_KON_MOD_Muurkap (Zincorplaat)		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement015	0CQpf9SY1DgOU1hBpiIM	66_KON_MOD_42306675-lift 2:	66_KON_MOD_42306675-lift 2		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement016	0CQpf9SY1DgOU1hBpiIL	66_KON_MOD_CWT_guiderail:6	66_KON_MOD_CWT_guiderail		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement017	0CQpf9SY1DgOU1hBpiIK	66_KON_MOD_CWT_guiderail:6	66_KON_MOD_CWT_guiderail		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement018	0CQpf9SY1DgOU1hBpiIJ	66_KON_MOD_Car_guiderail:66	66_KON_MOD_Car_guiderail		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement019	0CQpf9SY1DgOU1hBpiII	66_KON_MOD_Car_guiderail:66	66_KON_MOD_Car_guiderail		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	
http://www.schiphol.nl/ontology#IfcTransportElement020	0CQpf9SY1DgOU1hBpiIH	66_KON_MOD_Car:66_KON_MO	66_KON_MOD_Car		Schiphol parkeergarage P3	Parkeergarage P3	00 begane grond	

Figure 30: Example of a part of the validation report

Besides checking if a value is not empty, the application will be checking the value on correctness. A light way to check if a value can be right, is to check whether this value has the correct datatype. The datatype of a value can be indicated, for example, as a string, a boolean or a number. Besides checking if the datatype is correct, a check can be done if the value is realistic. For example, it will be unrealistic if a value is 0. When a cell has a value, but is selected by the light validation, the cell will be marked orange. This check will be determined according to the specific attribute or property, because these values have different constraints. In table 9 the attribute or property is shown with the corresponding constraint(s).

Table 9: Constraints of attributes and properties

Attribute or property	**Datatype constraint**	**Additional constraint**
GlobalId	String	Max of 22 digits/letters in the string
Name	String	
ObjectType	String	
STD	Float	2 digits, one dot, 2 digits Should be one of the codes of the STD list
Project	String	
Building	String	
Storey	String	Verdieping/level + a number
Space	String	
Clearwidth	Float	
ClearHeight	Float	
ClearDepth	Float	

These constraints are assimilated in the application which will result in a validation report. The empty values are shown red and the data constraint errors are shown orange. In paragraph 5.3.2 this is tested with the help of different models.

5.3.2 Validation of the proof of concept

To check if the application is working correctly for different IFC models a validation of the tool is executed. This validation has been done with the help of various models which are provided by Amsterdam Airport Schiphol. The models are from different projects and are modelled by different contractors or advisors as well. Some of the models are also relatively large. Running larger models in the application can need some loading time. To speed up the loading process of the application, only the elements which are needed to be checked are saved in a list. Because this is done first, the application does not have to run over all the elements in the model again every time a new property or attribute is asked for. The validation reports and results are explained per model.

- **KLM-ICA lounge**

The first project at Schiphol which was using BIM throughout the whole process is the KLM-ICA Lounge. Schiphol entered a procurement together with KLM to redevelop the current business lounge of KLM. This lounge is situated between the E-pier and the F-pier. The lounge will be extended with a third floor and an outside terrace. Besides this the lounge will be modernized and refurbished. Because Schiphol stated to fully implement BIM in the future, the BIM models of the project are demanded by Schiphol. This means that after the design phase different models were handed over. Currently, the contractor is constructing the project and the as-built models will be transferred to Schiphol after this phase.

To validate the application, the model of the subcontractor Kone is used. Kone is the supplier of the elevators for this project. The results of the application are a RDF file with the information as linked open data, according to the UDO, and an excel file with a validation report. The RDF file can be read in appendix VII. Only a part of this file is given in the appendix, due to the extended results. In the total report 81 transport elements are indicated. The transport elements are represented in this file in the following format;

```
aas:IfcTransportElement001 a aas:Elevator ;
   aas:capacitynr “0.0”^^xsd:float ;
   aas:capacitywght “0.0”^^xsd:float ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_M”^^xsd:unicode ;
   aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
      aas:hasName “66_KON_MOD_MACHINE_MX14_MONO:66_KON_MOD_MACHINE_MX14_MONO:416739”
^^xsd:string ;
   aas:hasSTD “”^^xsd:number ;
   aas:hasType “66_KON_MOD_MACHINE_MX14_MONO”^^xsd:string ;
   aas:project “ICA Lounge Schiphol”^^xsd:string ;
   aas:requires “None”^^xsd:string .
```

This representation tells that the transport element has the instance type of an elevator. Because all 81 elements have this instance, they are all categorized in this element class. The model actually does not include 81 elevators, but 3 elevators and 2 escalators. According to the name or the type of the element an indication can be made about what the object in reality is. To better categorize the different elements the instance should be indicated in the IFC model. When this is done for the whole object, the elevator which is a ifcTransportElement, the subclasses of the elevator, such as the cabin and counterweight can be indicated in the ontology as well. This way, the objects can be classified as an asset or an element part of an asset. A screenshot of the model can be seen in figure 31.

The second deliverable of the application is the validation report. This report indicates all the required attributes and properties and visualizes if a value is missing or incorrect. This report indicates that for all elements the STD/NL-SfB classification code is missing. Furthermore, he building and space the objects should be located in, are missing for all elements as well. The measurement quantities are found for some objects, but most values are missing as well. Depending on the actual element, this can be relevant or not. It is for example important to know the sizes of the whole elevator or of the elevator cabin but not of some small elements included in the elevator. When the elements are modelled better in the IFC this distinction can be improved.

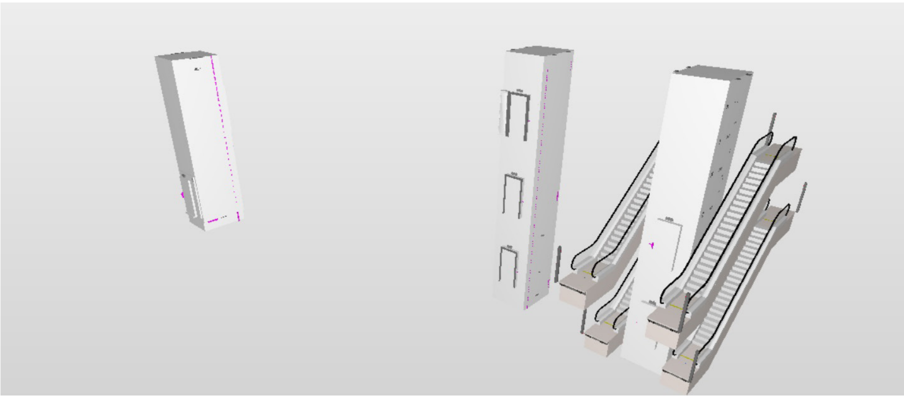


Figure 31: Screenshot of the KLM-ICA model in Solibri

- **Parking Garage 3**

The second project used for the validation of the application is the parking garage 3(P3). For this project, multiple models were made including the architectural model and the model with the elevators of the subcontractor Kone.

Because the model with the elevators is modelled by the same supplier, the application gives almost the same results as is extracted from the model of the KLM-ICA lounge indicated in the previous paragraph. The difference between the models is that the elevator model of the parking garage also includes some elevator doors, which are modelled as IfcDoor instead of IfcTransportElement. These elements always include a width and height in comparison to the transport elements. Besides this, the building is indicated in this project and this provides a better result in the validation report. A partial RDF file of this model can be found in appendix VII as well.

Not only the elevator model, but also the architectural model is validated by the application. This model includes many more objects but only the doors, wall, curtainwalls and transport elements are extracted and checked. The RDF file of this model indicates more attributes and properties. Of almost all elements, one of the following properties is found; is external, fire rating, load bearing or acoustic rating. Besides this, the model includes the STD codes, which are validated numbers. The validation report however indicates that no building name or spaces are found as well. A screenshot of both the models can be seen in figure 32.

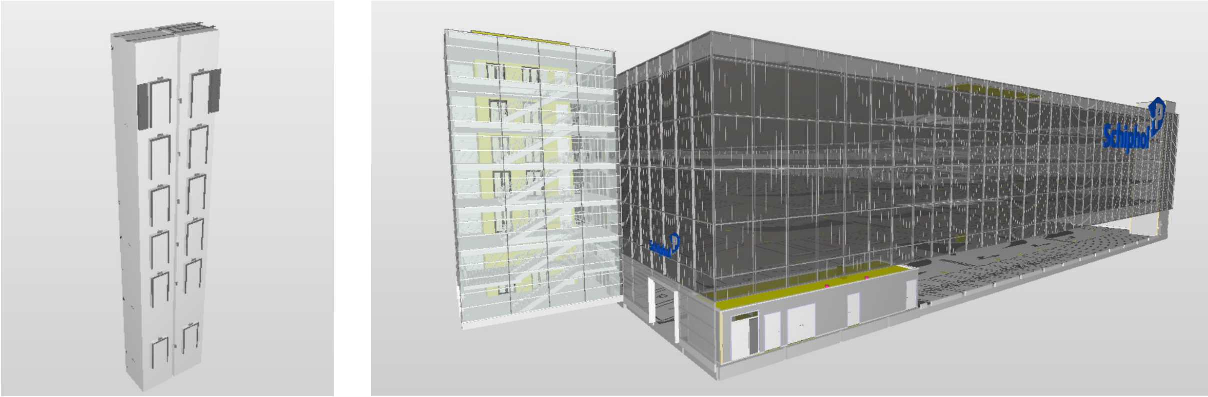


Figure 32: Screenshot of the elevator model (left) and the architectural model (right) of P3 in Solbiri

- **C-pier**

The third project used for the validation of the application is the C-pier. Only one model is used, which includes almost all aspects of the building. This model is made according to old models, drawings and 3D scans of the facilities to indicate the as-is situation of the building. The as-is situation will be used with the remodeling of the building. The new situation will be modelled in BIM as well.

The results do not differ much from the previous two projects. The partial RDF file can be found in appendix VII. In this project the STD codes and the spaces are missing as well. One main finding of the validation report is concerning the transport elements. In this model no elevators are found, in addition moving walkways are marked. This indicates that the instance type of the objects are used correctly in the BIM model. In the validation report, however, the difference cannot be seen immediately. The objects are all named ifcTransportElement, and this is not indicated in the validation report. In addition, this can be extracted from the RDF file. If you look at the name or type of the objects, this can be extracted, but this is dependent on the modeler of the BIM and the clearness of the names in the model. The picture of the model can be seen in figure 33.

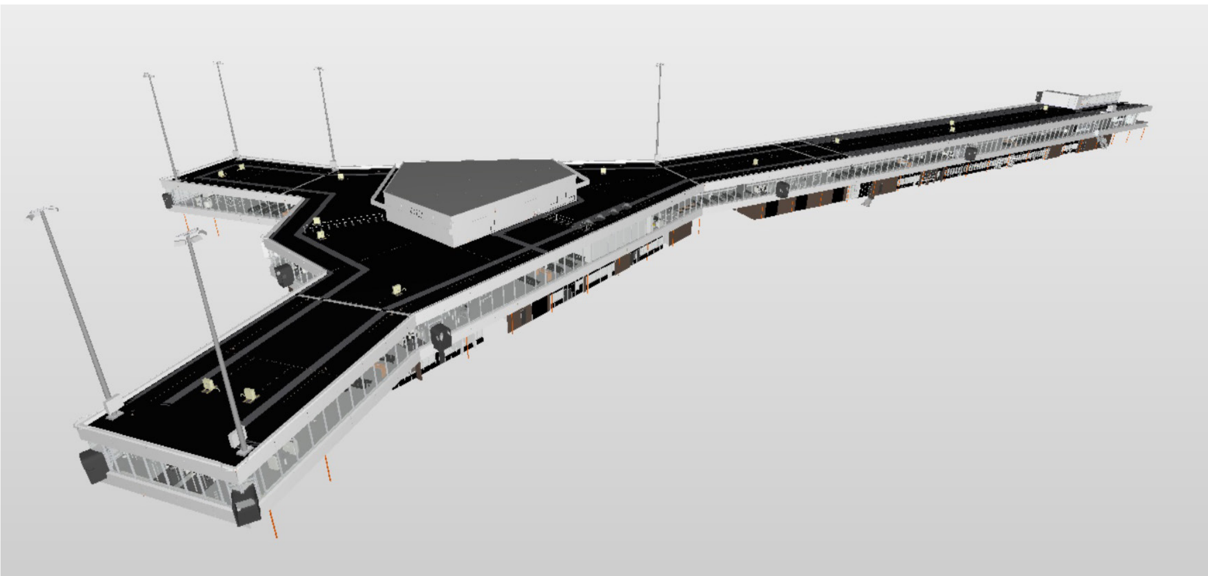


Figure 33: Screenshot of the model of the C-pier from Solibri

5.3.3 Conclusions of the proof of concept

From the three projects which are tested in the application, a conclusion can be drawn about the validation of the proof of concept. First of all, all models did get a result from the tool. This means that the application can be used independently of the model. The results of the models are comparable, so the application works in the same way for every model.

Secondly, the same mistakes or open values can be found in all the models. This means that the same values are not included in the IFC models or are not in the correct field of the model. This is something that can be improved in the future. When this is more clearly communicated with the modeler or when this is more clearly stated in the ILS of Schiphol, the values should be present correctly from now onwards. Another possibility is that the values can be found in the IFC model, but in a different field than expected. If this occurs often in the same way, the identification of these fields can be added to the application. For example, the STD code can be modelled as an ifcpropertysinglevalue, or as an ifcclassificationreference. The second value can be added to the application as well, to get a better result of the application in the future. Besides given empty values for the STD code, the application gives sometimes empty values for the building. This is because the building is often just modelled as an IfcBuilding, and no specific name is indicated. If no name is indicated this field stays empty. The application could not find the spaces in the mode, which indicates that no spaces are existing in the model. This does not have to be true, because sometimes there are spaces present in the model. The modeler just did not make the relation between the objects and the space. In this way the objects cannot be placed in the rooms.

However, because of the linked open data structure, there is a potential in using the application for finding indirect relations between objects, properties and attributes with queries. This can give opportunities for other solutions, when the information is not always modelled in the correct fields or place in the IFC models.

Which is notable, is that the tool also uses a validation of the datatype but no errors of this type are found in the models and marked orange. This means that the datatype is always used correctly. Besides checking the datatype, some constraints are taken into account as well. For example, for the STD code the classification code should be one of the values of the total list of STD codes. Because in the 3 projects no STD code is found, this constraint is not tested. When the model includes the STD codes, the change of errors will be larger and will give a better validated result.

Lastly, the tool is proposed as a proof of concept and is not a recommended user-friendly and developed application. Because of the immatureness of the application, some developments for improvement of the application are indicated in chapter 6.2, the discussion and future recommendations.

5.4 Business value

For the implementation of BIM in a company structure different aspects are important to take into account. Many people think that BIM can be implemented by only changing the information technology used in the company, but this is not the case. Introducing BIM in an organization means working with new structures where digital asset management will be central. This will change besides the software and hardware solutions, the working processes, and the management and organization of the company. Lastly, this will be influencing the human working culture in the company, because people should be getting used to this new way of working and adaption of the strategies used. The implementation factors are shown in figure 34 (Bouw Informatie Raad, 2015).

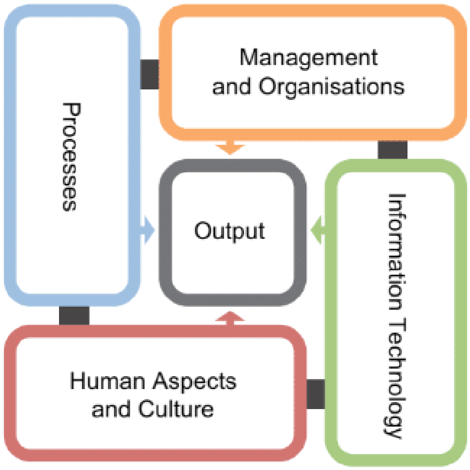


Figure 34: implentation factors of BIM

Because an organization has to put much effort in all the changes to implement BIM, this should have an added value to the business in return. To determine the added value and the way to capture this value during the new process implementation, this will be explained according to an analyzation of the value proposition after the implementation at Amsterdam Airport Schiphol.

One of the main jobs at an airport and also of AAS is maintenance during the lifecycle of the buildings. During this job, different 'pains' are found based on working with assets at airports. These pains can be divided in the following points;

- Working with continuous operations
- A lack of information
- Not knowing what will happen next

After using structured, uniform and validated information for asset management the following 'gains' can be reached;

- Predictability
- Better use of (existing) information
- A better exchange of information

To eliminate the pains and reach the gains, 3D asset/ information management should be used and a focus should be given to the quality and completeness of the information. When the information used by Schiphol can reach the correct quality, a better use of this information can be implemented in the future. This will give the opportunity of scenario planning, be more predictable and using real time data where quality and accurateness will be certain. In the end this will result in a shift from descriptive data analytics to predictive data maintenance.

When the new information exchange structure as explained in this thesis will be implemented, it will help with reaching the gains and with the shift to predictive maintenance. When all BIM implementation factors are taken into account, linked open data will provide the business with better information, which is more flexible and will give more insights in the information, processes and (possible) scenarios. When linked open data is used, this will improve the exchange of information between the different management systems and can make new relations between applications or interfaces possible. Besides this it can help improving the validation process to reach the needed quality of the information. When this needed quality is reached, semantics can help find relations between the data as well.

The implementation factors will ask for a change of internal processes and a better awareness in the organization, which will ask an adaptive culture change from its employees. To support the employees and process changes the management of the organization should be supporting the innovations and adapt their business vision and mission. Lastly, improved IT is needed to implement the innovation, which will include new hardware and software.

This will need for an initial investment, which will be regained by saving working hours and labour costs when the new and more efficient processes are implemented. Value will be added to the process, which will be optimized. Besides this, it will save costs during maintenance and operations spent on, for example, repairing or replacing assets for the airport.

As has been stated in chapter 1, the adoption of BIM is shifting from the early adopters to the late majority which indicates the importance of stepping in at this moment. When the market will be developed fully yet, it will get more difficult to have influence on the commonly used standards and working methods in the industry. Because Schiphol is a large asset management company which has many contractors and advisors working for them, they have a large influence on current innovations. For them it will be important to start implementing and standardizing their new working process now. Especially if they want to determine how the market will implement this from now onwards.

6. Conclusion and discussion

This chapter of the report will present the conclusion of the overall thesis. The conclusion will provide answers to the research questions. In this chapter the main objectives of the report will be discussed, but also some limitations will be given in the discussion. The discussion will add some future recommendations to elaborate on the thesis and complement the research from now onwards.

6.1 Conclusion

The conclusions are drawn according to the research questions which are stated in paragraph 2.2. The main focus of the research is put on transferring and validating information from the construction projects to asset management by using BIM. By using validated digital information, asset management processes can be improved. The improvement of asset management processes will result in more efficient maintenance and operations of the assets, with less failure costs and better insights. Better insights will lead to more predictability in the future.

The main research question of the thesis is; **"How to transfer and validate as-built information to capture added value for digital asset management?"**. The main research question can be answered based on the conclusions drawn from the sub-questions of the thesis. Therefore, the five sub-questions will be answered first;

Which information does an asset manager need for digital maintenance and should be handed over after the construction phase?

Based on the conducted study, the asset manager always needs a combination of a graphical model, which has non-graphical information and documents attached to it. Because maintenance is always focused on assets, this information should be stored object-based. For asset management, the non-graphical information will be of the most importance. Every asset should comply to the core data requirements, the business requirements and the specific asset requirements. To fulfill these requirements the data should at least be complete, heterogeneous, consistent, reliable, usable, secure and have added value. Besides this, the following aspects should be provided by the non-graphical information of an asset;

- The information to identify the asset
- The location of the asset
- The status of the asset
- The maintenance records of the asset
- The documents needed for legal obligations

This information will be defined in the UDO of the asset manager and should be handed over according to open standards. If the basic characteristics of every asset are correctly modelled in the BIM, this will lead to much more efficient asset management.

How should this information be handed over to asset manager according to open standards?

The handover of the information is an important moment in the whole building process after the realization of the building. Different sorts of information exchange structures exist, which are based on open standards. To provide for a sufficient standard which can be extended and be aligned to other objects apart from building objects, the best choice is to use a linked open data framework. This structure is object-based and can be easily extended or aligned with other ontologies. The handover of the information should be exchanged based on semantics. Before this can be implemented, the asset manager should be able to process the linked open data into their management systems. Therefore, the main part of this research investigates and provides a proof of concept of how to use an UDO to transfer information. The information will be transferred from the IFC file to an RDF file according to the predefined information structure in the ontology. The RDF file can be constructed with the application, which indicates the potential of transferring information from an IFC file into RDF. After the RDF file is constructed, the information can be queried and relations between objects, attributes and properties can be found and indicated. To relate documents to the information structure and provide in an overall container the ICDD framework can be potentially used in the future. This container can transfer BIM models, with non-graphical information and object-based attached documents.

How can the asset manager validate this as-built information or how can the contractor suffice on the requirements set by the company?

The validation of the information is important to guarantee the quality of the data. If no validation is used, no real assumptions can be done and this will be a crucial step towards improving asset management processes. Based on the linked open data structure a validation can be executed on the as-built information. The information can be checked quantitative, this means that the validation checks if all required information is present in the data structure. Besides this, a light qualitative check can be applied to the validation, which will check the information on data types. All this can be done automatically and is implemented in the proof of concept. The application indicates that data types are mostly used correctly, but also which information is missing in the files, or is not found in the correct place. This can be communicated to the contractor who can improve the information in the IFC files. Besides this, the information should be checked on content as well. This check will verify if the information is correct according to the project. This should be done by a person included in the project and should be done manually.

How can the company process this as-built information into their own asset management systems?

After the validation of the information in the process, the data should be transferred into the asset management system. The information required is dependent on the system. A separation can be made between the systems. The asset management system only stores asset information, the GIS system stores geometry, which is geographical information and lastly, a document management system can store documents.

The validated asset information can be transferred from the excel file to the asset management system easily. When this information is stored in a common data environment (CDE) or a data hub, the connection between the asset management system and CDE can be made. When the CDE is object-based and includes linked open data, then interfaces between all systems and other (future) applications can be realized.

What is the added value for asset management and how can this value be captured during this process?

When the information structure based on linked open data can be implemented in the asset management processes, it will provide easier usage and will maintain the information about the assets. By using the UDO to validate and store information, the as-built data of building assets as well as data of other assets can be transferred to the CDE according to this structure. When this system will be object-based and provide validated information, a real 'one-single source of truth' can be provided to the asset manager. This will result in better insights in the information, easier exchange of information and more efficient maintenance, which will add value to the processes for the asset manager. By using linked open data, an easier and automatic transfer and validation process can be established, which will save time and costs. To reach this, an investment is required which includes investing in IT, new processes, the management and organizations' vision and the human culture at the organization, but at the same time it will capture added value in the future.

The conclusions from the above sub-questions contribute to the answer of the main research question;

How to transfer and validate as-built information to capture added value for digital asset management?

To transfer, validate and use as-built information for digital asset management, linked open data will currently provide the best opportunity to reach this, based on an open standard. By defining an UDO based on the requirements of the asset manager, an information structure can be defined, which gives the opportunity to validate the information according to this structure. An ontology has the benefit that it can be easily extended and aligned with other ontologies which, for example, can include information that cannot be extracted from a building model. Besides this, the ontology provides an object-based structure, which will be the second benefit. With the use of semantics, the as-built information can be automatically validated based on completeness and correctness of the data and only a manual project-based validation will be needed. This structure will promote many opportunities to improve asset management and will lead to predictive maintenance in the future. Implementing these processes for digital asset management will capture added value for the company.

6.2 Discussion and future recommendations

After the conclusions are drawn from this thesis, the results will be discussed and points of improvement for the future will be given in the form of recommendations.

This thesis recommends to use linked open data as an information structure to transfer and validate as-built information. The use of linked open data will give many potentials, however not many implementation use cases are currently given in the industry, because linked open data is still under development. Additional research and development should be done before it is fully ready to use and implement in the business. This thesis provides the first steps to implement this structure, but some additional research should be put into transferring information between stakeholders. In this report, a suggestion is given to use the ICDD, which gives the potential to use an linked data structure to combine geometrical, not-geometrical information and documents in an object-based container.

Secondly, an information structure is provided for the environment and an explanation is given about how to extent the ontology. The extension of the ontology should be researched and tested according to the requirements of all assets. The information environment where the linked data will be stored, is of importance as well. This CDE or data hub should be the one single source of truth from now onwards. It is important to state what information is from which source and if the information is validated. The exchange interfaces between the different applications or systems and the information environment should be researched and the information flows between them should be defined.

Thirdly, a proof of concept is provided with an application. This application however is immature, but because no other software solutions exist, it can be used to transfer IFC files to RDF files and to validate the information in the RDF. This application can be improved by adding more assets and requirements as well. Besides this, the output of the application can be improved. Currently, an Excel file is generated, which is useful for the asset management systems. However, when the information quality should be improved, a better solution can be to work with the open standard BCF to communicate the issues with the contractor who is responsible for the model. A BCF file can be generated from the application by adding an XML output generator. The BCF file has an open standard file format and the information is attached to a GUID of an object. To implement this to the application, a viewer of the model is needed as well. This viewer can provide the screenshot which is needed, as an png file attached in the XML file of the BCF.

The improvement of the research and the development of the information structure and information environment, can have many potentials for asset management according to open standards in the future.

7. References

Badrinath, A. C., Chang, Y., & Hsieh, S. (2016). An Overview of Global Research Trend in BIM from Analysis of BIM Publications, (September).

Becerik-Gerber, B., Jazizadeh, F., Li, N., & Calis, G. (2012). Application areas and data requirements for BIM-enabled facilities management. Journal of Construction, (March), 431–442. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000433.

Beetz, J. (2016). Industry Foundation Classes: an introduction. Course 7M900. Week 5, slide 10. Retrieved on 10-6-2017.

Beetz, J., van Leeuwen, J., & de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23(1), 89. https://doi.org/10.1017/S0890060409000122

Berlo, L. Van. (2010). IFC-based clash detection for the open-source BIMserver P van den Helm, M Böhms & L van Berlo Survey of known technologies used for IFC clash detection.

Berlotti. (2012). IFD: International Framework for Dictionaries – ISO 12006-3. Nationaal BIM handbook. BuildingSMART standards. Retrieved on 9-10-2017, from http://nationaalbimhandboek.nl/woordenboek/ifd-international-framework-for-dictionaries-iso-12006-3/

Berners-Lee, T. Linked Data — Design Issues, 2006, Retrieved September 15, 2017, from http://www.w3.org/DesignIssues/LinkedData.html.

Berners-Lee, T., Shadbolt, N., & Hall, W. (2006). The Semantic Web Revisited. IEEE Computer Science.

Borrmann, A., Beetz, J., Koch, C., & Liebich, T. (2015). Building Information Modeling: Technologische Grundlagen und industrielle Praxis. Chapter 6 Industry Foundation Classes – A vendor-independent data model for the entire life cycle of a building. Springer.

Bouw Informatie Raad. (2014). BIR Kenniskaart nr.1: Nederlandse BIM Levels, (1), 2. Retrieved from http://www.bouwinformatieraad.nl/wp-content/uploads/2014/10/kaart01-ned.pdf
Bouw Informatie Raad. (2015). BIR Kenniskaart nr. 0 Wat is BIM?, (September), 2. Retrieved from www.bouwinformatieraad.nl

Bsi. (2013). Specification for information management for the capital/delivery phase of construction projects using building information modelling: PAS 1192-2:2013. BSI Standards Publication, (1), 1–68. https://doi.org/Published by the British Standard Institute. British Standard Limited. ISSN9780580781360. /BIM TASK GROUP

BSI. (2014a). BS 1192-4 : 2014 Collaborative production of information Part 4 : Fulfilling employer' s information exchange requirements using COBie- Code of practice. British Standards Institution (BSI), 58. Retrieved from http://shop.bsigroup.com/forms/BS-1192-4/

BSI. (2014b). PAS 1192-3:2014 Specification for information management for the operational phase of assets using building information modelling. BSI Standards Publication, 1(March), 1–44. https://doi.org/Published by the British Standard Institute. British Standard Limited. ISSN9780580781360. /BIM TASK GROUP

Building smart. (2010). Information Delivery Manual Guide to Components and Development Methods. buildingSMART, 1–84.

BuildingSMART. (2012). Technical vision: openBIM. Retrieved September 10, 2017, from http://www.buildingsmart.org/standards/technical-vision/

BuildingSMART. (2018). IfcElement. Retrieved Februari 18, 2018, from http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcproductextension/lexical/ifcelement.htm

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., & O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. Advanced Engineering Informatics, 27(2), 206–219. https://doi.org/10.1016/j.aei.2012.10.003

Dakhil, A., Alshawi, M., & Underwood, J. (2015). BIM Client Maturity: Literature Review, (June), 229–238.

East, W. E., & Brodt, W. (2007). BIM for Construction Handover. Journal of Building Information Modeling, 28–35. Retrieved from https://www.pinnaclecad.com/pin-pdf/BIM-for-Construction-Handover.pdf

Eastman, C., Lee, J. min, Jeong, Y. suk, & Lee, J. kook. (2009). Automatic rule-based checking of building designs. Automation in Construction, 18(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Eastman, C. M., Jeong, Y., Sacks, R., & Kaner, I. (2010). Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards, 24(February), 25–34.

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2008). BIM handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Hoboken, NJ: Wile.

Farias, T. M., Roxin, A., & Nicolle, C. (2015). COBieOWL, an OWL Ontology Based on COBie Standard, 9415, 361–377. https://doi.org/10.1007/978-3-319-26148-5

Farias, T. M. De, Roxin, A.-M., & Nicolle, C. (2015). IfcWoD , Semantically Adapting IFC Model Relations into OWL Properties. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands, 175–185.

Fernal Berrer, F. J. (2017). Internal design validation attestation in BIM models- A combination of Semantic Web technologies and BIM Collaboration Format, 185.

Gruber, T. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199-220.

Hoeber, H., & Alsem, D. (2016). Engineering , Construction and Architectural Management: Life-cycle information management using open-standard BIM. Department of Planning and Transport, Royal HaskoningDHV, Nijmegen, The Netherlands.

Hoeber, H., Alsem, D., & Willems, P. (2015). The management of information over the life-cycle of a construction project using open-standard BIM. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands, 295–301.

Ifma, edited by P. T. (2013). BIM for Facility Managers. (P. Teichholz, Ed.) (1st ed.). John Wiley & Sons Inc.

International Organization for Standardization, I. (2017a). ISO NP 21597-1 Information container for Data Drop- Exchange specification- Part 1: Container (Draft version).

International Organization for Standardization, I. (2017b). ISO NP 21597-2 Information container for Data Drop- Exchange specification- Part 2: Dynamic semantics (Draft version).

Johnston, J. (2013). COBie explained, (May).

Love, P. E. D., Matthews, J., Simpson, I., Hill, A., & Olatunji, O. A. (2014). A benefits realization management building information modeling framework for asset owners. Automation in Construction, 37, 1–10. https://doi.org/10.1016/j.autcon.2013.09.007

Lu, W., Fung, A., Peng, Y., Liang, C., & Rowlinson, S. (2015). Demystifying construction project time-effort distribution curves: a BIM and non-BIM comparison, (February 2016).

Motik, B.C. Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz. (2012). OWL2 web ontology language reference profiles (second edition). W3C recommendation 11 December 2012. Retrieved September 29, 2017, from https://www.w3.org/TR/owl2-overview/

NBS. (2017). NBS National BIM Report 2017. What are the key findings in the 2017 report? Retrieved October 6, 2017, from https://www.thenbs.com/knowledge/nbs-national-bim-report-2017

Nederveen, S. van, Beheshti, R., & Willems, P. (2010). Building Information Modelling in the Netherlands: A Status Report. In W078-Special Track 18th CIB World Building Congress. Salford, United Kingdom.

O'Keeffe, A., Alsem, D., Corbally, R., van Lanen, R., & Interlink. (2017). Information Management for European Road Infrastructure using Linked Data. Investigating the Requirements.

Oh, M., Lee, J., Hong, S. W., & Jeong, Y. (2015). Integrated system for BIM-based collaborative design. Automation in Construction, 58, 196–206. https://doi.org/10.1016/j.autcon.2015.07.015

Out-law. (2012). Out-law.com, Legal news and guidance from Pinsent Masons. Building Information Modelling. Retrieved September 25, 2017, from https://www.out-law.com/en/topics/projects--construction/projects-and-procurement/building-information-modelling/

Paasiala, P., Laukala, J., Lifländer, L., Linhard, K., Pijnenburg, E., Berlo, L. (2017). BIM Collaboration Format v2.1 Technical Documentation. Retrieved December 15, 2017, from https://github.com/BuildingSMART/BCF-XML/tree/master/Documentation

Pandit, S. (2017). What is the significance (or meaning) of Tbox and Abox in semantic web?. Retrieved November 1, 2017, from https://www.quora.com/What-is-the-significance-or-meaning-of-Tbox-and-Abox-in-semantic-web

Patacas, J., Dawood, N., Vukovic, V., & Kassem, M. (2015). BIM for facilities management: Evaluating BIM standards in asset register creation and service life planning. Journal of Information Technology in Construction, 20(January), 313–331.

Pauwels, P., & Roxin, A. (2016). SimpleBIM : From full ifcOWL graphs to simplified building graphs. 11th European Conference on Product and Process Modelling, (October), 11–18.

Pauwels, P., Schneider, G. F., Rasmussen, M. H., Lefrancois, M., Hviid, C. A., & Karlshøj, J. (2017). Recent changes in the Building Topology Ontology, (October).

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction, 63, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. Automation in Construction, 73, 145–165. https://doi.org/10.1016/j.autcon.2016.10.003

Prud'hommeaux, E., Seaborne, A. (2008). SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. Retrieved December 19, 2017 from https://www.w3.org/TR/rdf-sparql-query/

Rasmussen, M. H., Pauwels, P., Hviid, C. A., & Karlshøj, J. (2017). Proposing a central AEC Ontology that allows for domain specific extensions. LC3 3017, (Volume I-Proceedings of the Joint Conference on Computing in Construction), 237–244.

Royal Institute of British Architects. (2013). RIBA plan of work 2013. RIBA: London, 6, 1–27. https://doi.org/ISBN 978 1 85946 519 6

Sack, H. (2015). Knowledge engineering with Semantic Web Technologies. Lecture 1: Knowledge Engineering and the web of data. 1.3 Understanding content on the Web. Hasso-Plattner-Institut for IT Systems Engineering. University of Potsdam.

Schiphol Group. (2017). SCHIPHOL: NEGEN PROCENT MEER PASSAGIERS IN 2016. Retrieved September 18, 2017, from https://www.luchtvaartnieuws.nl/nieuws/categorie/3/airports/schiphol-negen-procent-meer-passagiers-in-2016

Sebastian, R., & van Berlo, L. (2010). Tool for Benchmarking BIM Performance of Design, Engineering and Construction Firms in The Netherlands. Architectural Engineering and Design Management, 6(4), 254–263. https://doi.org/10.3763/aedm.2010.IDDS3

Spence, R., & Mulligan, H. (1995). Sustainable development and the construction industry. Habitat International, 19(3), 279–292. https://doi.org/10.1016/0197-3975(94)00071-9

Stancheva, M. (2017). Improving the management of structural engineering requirements in the design phase. Linking project requirements to BIM, based on the Semantic Web and Linked Data principles. Eindhoven, University of Technology.

Top, J.L. (2017). Wageningse Ontology of Measurements komt als beste uit de bus. Nieuws, 14 augustus 2017. Retrieved December 19, 2017 from https://www.wur.nl/nl/nieuws/Wageningse-Ontology-of-Measurements-komt-als-beste-uit-de-bus.htm

Stichting Stabu. (2016). Het specificeren van de bouwwerkinformatie-inhoud. Opdracht B&U, Deelopdracht 1, in opdracht van Rijksvastgoedbedrijf, Schiphol, Erasmus MC.

Törmä, S., Oraskari, J., & Hoang, N. (2012). Distributed transactional building information management. Proceedings of the 1st Workshop Linked Data in Architecture and Construction (LDAC2012). Retrieved from http://multimedialab.elis.ugent.be/ldac2012/documents/LDACworkshopreport.pdf#page=12

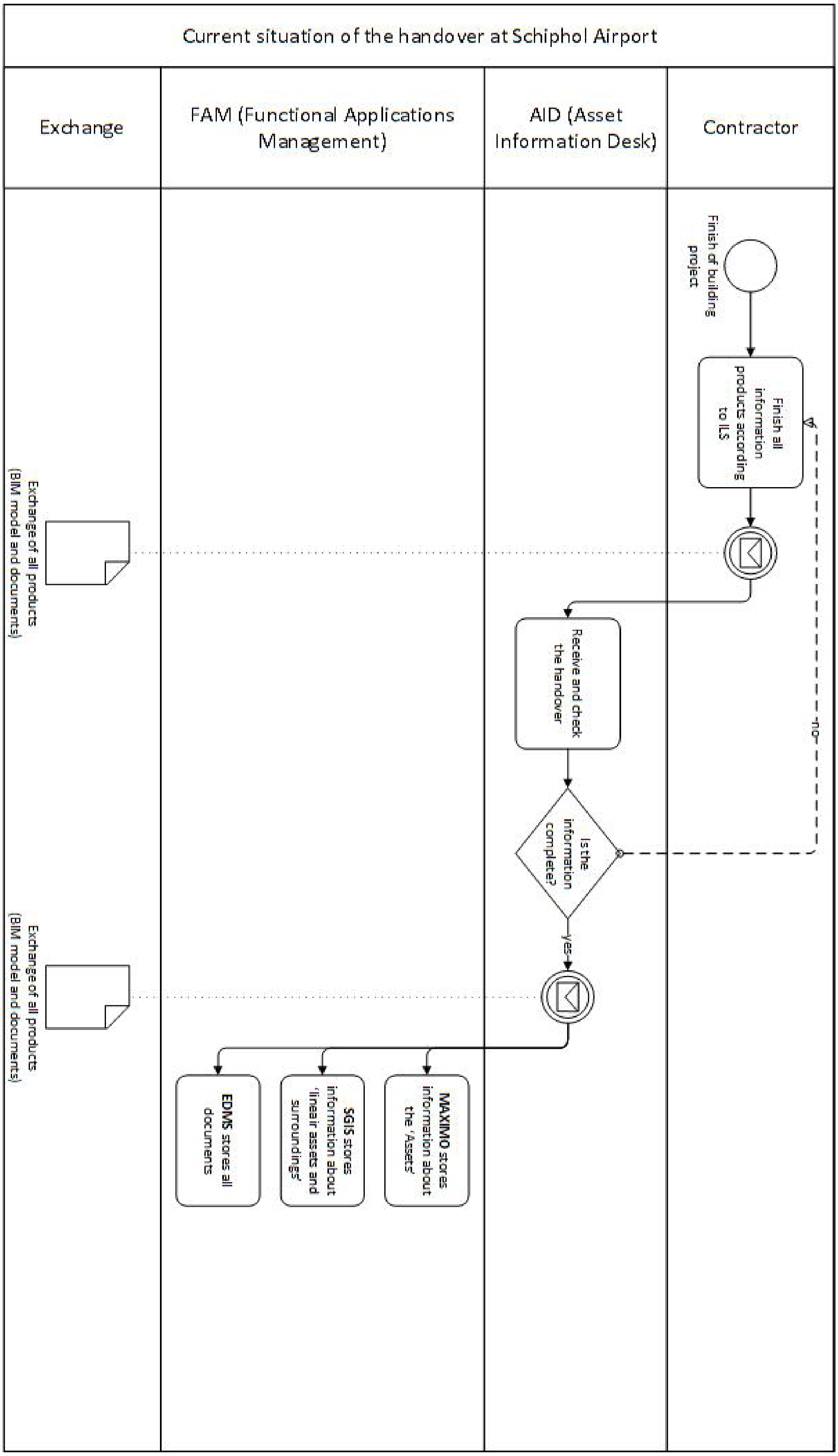
Willems, P. (2008). Beschrijving uitwisselingsformaat – COINS Container. COINSweb. Retrieved October 2, 2017 from http://www.coinsweb.nl/wiki/index.php/Beschrijving_uitwisselingsformaat_%E2%80%93_COINS_Container

Wood, B. (2012). COBie Data Drops - Structure, uses & examples. Bis, 1–24. https://doi.org/Publisher:Cabinet Office BIS department for Business Innovation&Skills. Uk Government.

Zhang, C., Beetz, J., & Weise, M. (2015). Interoperable validation for IFC building models using open standards. Journal of Information Technology in Construction, 20(December 2014), 24–39. https://doi.org/ISSN 1874-4753

Appendix I: BPMN Process map

BPMN Process map of the handover of a building project at Schiphol Airport



Appendix II: Interview questionnaire in Dutch

Doel van de interviews:
Behoefte van Schiphol in kaart brengen vanuit verschillende invalshoeken. Hiervoor zullen een aantal verschillende personen werkzaam bij Schiphol geïnterviewd worden. Aan deze personen wordt gevraagd hoe hun het ASM proces bekijken vanuit hun perspectief en wat hun rol hier in is. Daarnaast wordt er gevraagd naar wat hun visie/mening is t.o.v. BIM inzetten voor ASM in de toekomst. Wat zijn voor deze persoon de knelpunten/problemen op dit moment en hoe kan dit verbeterd worden?

Algemeen:
1. Wat is jouw functie binnen Schiphol en wat zijn jouw taken in het proces tijdens de bouw en overdracht van een nieuw gebouw? (Bekijken in relatie tot BPMN proces afbeelding).

2. Heb jij BIM (Building Information Management) al toegepast tijdens een project? Wat was jouw ervaring met BIM en wat kan er verbeterd worden in de toekomst?

3. Op wat voor manier kan naar jouw mening BIM een meerwaarde gaan toevoegen in jouw huidige werkproces?

Overdracht:
4. Is de informatie uitvraag zoals geformuleerd in de ILS 3.0/3.1 goed genoeg? Of is deze uitvraag te uitgebreid of juist te beperkt? Wat zou er weg moeten/bij moeten komen?

5. Hoe zie jij de rol van Schiphol tijdens de data drop/overdracht momenten? Zal de rol van Schiphol juist actief of passief moeten zijn?

6. Wordt op dit moment de informatie die wordt gevraagd over de assets compleet aangeleverd? Of ontbreekt er informatie? Wordt deze missende informatie dan door Schiphol zelf toegevoegd aan de beheersystemen?

7. Naast het BIM model en de informatie in dit model, blijft er een behoefte aan documenten (Afbeelding PAS 1192-2). Moet deze documenten apart aangeleverd nodig of is het juist beter als deze worden gelinkt aan het model?

Asset Management:
8. Als de informatie vanuit BIM naar het FM systeem automatisch kan worden overgezet, zou het beheerproces dan daarna hetzelfde moeten blijven? Of zie je hier ook nieuwe kansen in?

9. Zie je het als een mogelijkheid dat er op Schiphol een linked open database wordt gemaakt, waaraan verschillende systemen zoals MAXIMO en SGIS gekoppeld zijn? Zou dit voordelen opleveren voor jouw huidige werkmethodiek?

10. Wanneer alle nieuwe projecten in BIM worden gegaan, denk je dat de bestaande projecten dan ook moeten worden overgezet naar BIM?

Appendix III: UDO of Schiphol (.ttl)

```
@prefix : <http://www.schiphol.nl/ontology#> .
@prefix aas: <http://www.schiphol.nl/ontology#> .
@prefix ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://www.schiphol.nl/ontology> .

<http://www.schiphol.nl/ontology> rdf:type owl:Ontology .

#################################################################
#    Object Properties
#################################################################

###  http://www.schiphol.nl/ontology#IsContainedBy
aas:IsContainedBy rdf:type owl:ObjectProperty ;
            rdfs:subPropertyOf owl:topObjectProperty ;
            owl:inverseOf aas:contains ;
            rdfs:domain <https://w3id.org/bot#Zone> ;
            rdfs:range <https://w3id.org/bot#Zone> .

###  http://www.schiphol.nl/ontology#contains
aas:contains rdf:type owl:ObjectProperty ;
         rdfs:subPropertyOf owl:topObjectProperty ;
         rdfs:domain ifc:IfcSystem ,
                 <https://w3id.org/bot#Zone> ;
         rdfs:range ifc:IfcElement .

###  http://www.schiphol.nl/ontology#hosts
aas:hosts rdf:type owl:ObjectProperty ;
        rdfs:subPropertyOf owl:topObjectProperty ;
        rdfs:domain ifc:IfcElement ;
        rdfs:range ifc:IfcElement .

###  http://www.schiphol.nl/ontology#isRequiredBy
aas:isRequiredBy rdf:type owl:ObjectProperty ;
            rdfs:subPropertyOf owl:topObjectProperty ;
            owl:inverseOf aas:requires ;
            rdfs:domain aas:Eis .

###  http://www.schiphol.nl/ontology#realises
aas:realises rdf:type owl:ObjectProperty ;
         rdfs:domain ifc:IfcElement ;
         rdfs:range aas:Function .

###  http://www.schiphol.nl/ontology#requires
aas:requires rdf:type owl:ObjectProperty ;
         rdfs:subPropertyOf owl:topObjectProperty ;
         rdfs:domain ifc:IfcElement ;
         rdfs:range aas:Eis ,
                aas:Kenmerk .


#################################################################
```

```
#    Data properties
#################################################################

###  http://www.schiphol.nl/ontology#hasBoolean
aas:hasBoolean rdf:type owl:DatatypeProperty ;
          rdfs:range xsd:boolean .

###  http://www.schiphol.nl/ontology#hasString
aas:hasString rdf:type owl:DatatypeProperty ;
         rdfs:range xsd:string .

###  http://www.schiphol.nl/ontology#hasValue
aas:hasValue rdf:type owl:DatatypeProperty ;
         rdfs:range xsd:decimal ,
              xsd:double ,
              xsd:float ,
              xsd:int .


#################################################################
#    Classes
#################################################################

###  http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcBuildingElement
ifc:IfcBuildingElement rdf:type owl:Class ;
             rdfs:subClassOf ifc:IfcElement ;
             rdfs:label "BuildingElement"@en .

###  http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcCurtainWall
ifc:IfcCurtainWall rdf:type owl:Class ;
           rdfs:subClassOf ifc:IfcBuildingElement ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty aas:requires ;
                     owl:someValuesFrom aas:AcousticRating
                   ] ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty aas:requires ;
                     owl:someValuesFrom aas:Compartmentation
                   ] ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty aas:requires ;
                     owl:someValuesFrom aas:FireRating
                   ] ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty aas:requires ;
                     owl:someValuesFrom aas:IsExternal
                   ] ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty aas:requires ;
                     owl:someValuesFrom aas:LoadBearing
                   ] ;
           rdfs:label "CurtainWall"@en .

###  http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcDoor
ifc:IfcDoor rdf:type owl:Class ;
       rdfs:subClassOf ifc:IfcBuildingElement ,
               [ rdf:type owl:Restriction ;
                 owl:onProperty aas:requires ;
```

```
                owl:someValuesFrom aas:AcousticRating
               ] ,
               [ rdf:type owl:Restriction ;
                owl:onProperty aas:requires ;
                owl:someValuesFrom aas:FireRating
               ] ,
               [ rdf:type owl:Restriction ;
                owl:onProperty aas:requires ;
                owl:someValuesFrom aas:IsExternal
               ] ;
      rdfs:label "Door"@en .

###  http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcElement
ifc:IfcElement rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty aas:contains ;
                  owl:someValuesFrom ifc:IfcElement
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty aas:requires ;
                  owl:someValuesFrom ifc:IfcLabel
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty aas:requires ;
                  owl:someValuesFrom aas:Measurement
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty aas:requires ;
                  owl:someValuesFrom aas:Status
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty aas:requires ;
                  owl:someValuesFrom <http://www.schiphol.nl/ontology/ontology#Locatie>
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty aas:requires ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onClass ifc:IfcGloballyUniqueId
                 ] ;
        owl:disjointWith ifc:IfcSystem ,
                 aas:Building ,
                 aas:Site ,
                 aas:Space ,
                 aas:Storey ,
                 <https://w3id.org/bot#Zone> ;
        rdfs:label "Element" ,
             "Object"@en .

###  http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcGloballyUniqueId
ifc:IfcGloballyUniqueId rdf:type owl:Class ;
           rdfs:subClassOf aas:Kenmerk ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:hasString ;
                   owl:allValuesFrom xsd:string
                  ] ;
            rdfs:comment "An IfcGloballyUniqueId holds an encoded string identifier that is used to uniquely identify an IFC object. An IfcGloballyUniqueId is a Globally Unique Identifier (GUID) which is an auto-generated 128-bit number. Since this identifier is required for all IFC object instances, it is desirable to compress it to reduce
```

```
overhead."@en ;
                rdfs:label "GUID"@en .

### http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcLabel
ifc:IfcLabel rdf:type owl:Class ;
        rdfs:subClassOf aas:Kenmerk ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:hasString ;
                   owl:allValuesFrom xsd:string
                  ] ;
        rdfs:label "Name"@en .

### http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcSystem
ifc:IfcSystem rdf:type owl:Class ;
         rdfs:subClassOf aas:Aspect ;
         owl:disjointWith <https://w3id.org/bot#Zone> ;
         rdfs:label "System"@en .


### http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcTransportElement
ifc:IfcTransportElement rdf:type owl:Class ;
                rdfs:subClassOf ifc:IfcElement ;
                rdfs:label "TransportElement"@en .

### http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcWall
ifc:IfcWall rdf:type owl:Class ;
        rdfs:subClassOf ifc:IfcBuildingElement ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:requires ;
                   owl:someValuesFrom aas:AcousticRating
                  ] ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:requires ;
                   owl:someValuesFrom aas:Compartmentation
                  ] ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:requires ;
                   owl:someValuesFrom aas:FireRating
                  ] ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:requires ;
                   owl:someValuesFrom aas:IsExternal
                  ] ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty aas:requires ;
                   owl:someValuesFrom aas:LoadBearing
                  ] ;
        rdfs:label "Wall"@en .

### http://ifcowl.openbimstandards.org/IFC4_ADD2#objectType_IfcObject
ifc:objectType_IfcObject rdf:type owl:Class ;
                rdfs:subClassOf aas:Kenmerk ,
                        [ rdf:type owl:Restriction ;
                         owl:onProperty aas:hasString ;
                         owl:someValuesFrom xsd:string
                        ] ;
                rdfs:label "Type"@en .
```

```
###  http://www.schiphol.nl/ontology#AcousticRating
aas:AcousticRating rdf:type owl:Class ;
            rdfs:subClassOf aas:Kenmerk ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty aas:hasValue ;
                        owl:allValuesFrom xsd:int
                      ] .

###  http://www.schiphol.nl/ontology#Aspect
aas:Aspect rdf:type owl:Class ;
       rdfs:subClassOf owl:Thing .

###  http://www.schiphol.nl/ontology#Asset
aas:Asset rdf:type owl:Class ;
      rdfs:subClassOf aas:Aspect .

###  http://www.schiphol.nl/ontology#Building
aas:Building rdf:type owl:Class ;
        rdfs:subClassOf <https://w3id.org/bot#Zone> ;
        owl:disjointWith aas:Site ,
                  aas:Space ,
                  aas:Storey .

###  http://www.schiphol.nl/ontology#ClearDepth
aas:ClearDepth rdf:type owl:Class ;
         rdfs:subClassOf aas:Measurement .

###  http://www.schiphol.nl/ontology#ClearHeight
aas:ClearHeight rdf:type owl:Class ;
         rdfs:subClassOf aas:Measurement .

###  http://www.schiphol.nl/ontology#ClearWidth
aas:ClearWidth rdf:type owl:Class ;
         rdfs:subClassOf aas:Measurement .

###  http://www.schiphol.nl/ontology#Compartmentation
aas:Compartmentation rdf:type owl:Class ;
             rdfs:subClassOf aas:Kenmerk ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty aas:hasValue ;
                         owl:allValuesFrom xsd:boolean
                       ] .

###  http://www.schiphol.nl/ontology#Component
aas:Component rdf:type owl:Class ;
        rdfs:subClassOf aas:Aspect .

###  http://www.schiphol.nl/ontology#Demolish
aas:Demolish rdf:type owl:Class ;
        rdfs:subClassOf aas:Status .

###  http://www.schiphol.nl/ontology#Eis
aas:Eis rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
               owl:onProperty aas:isRequiredBy ;
               owl:someValuesFrom ifc:IfcElement
              ] ,
              [ rdf:type owl:Restriction ;
```

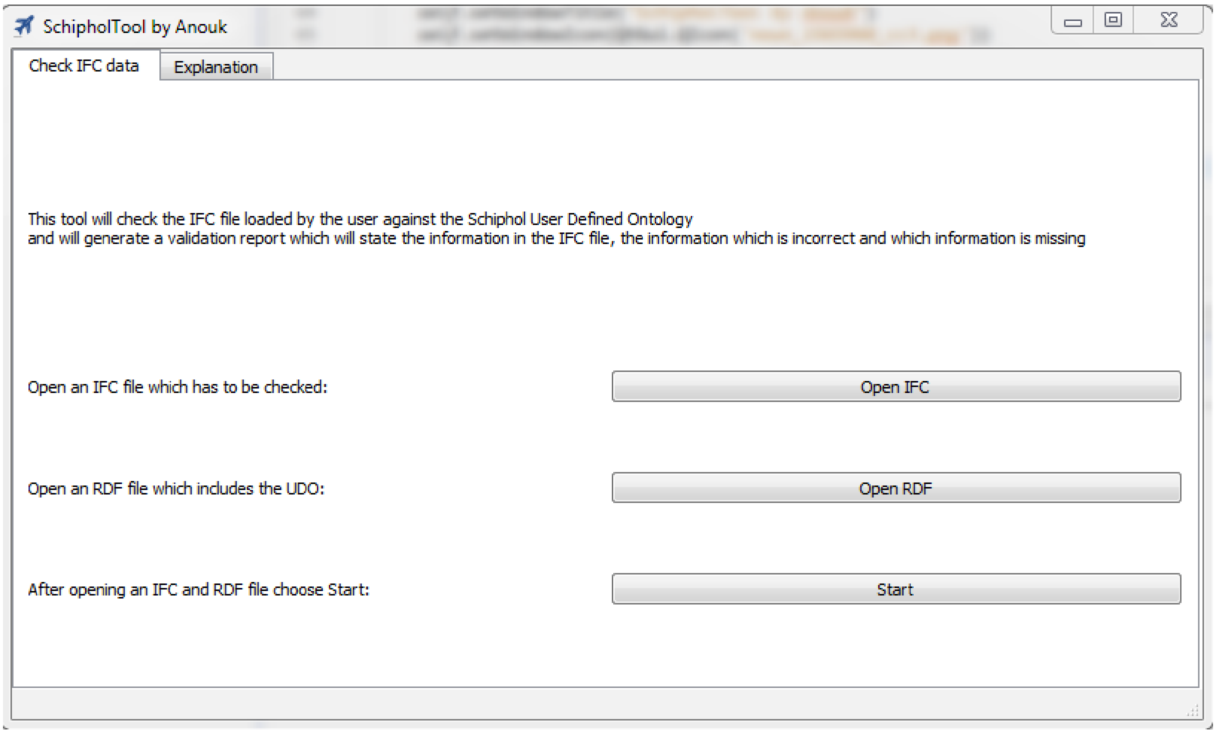
```
                    owl:onProperty aas:isRequiredBy ;
                    owl:someValuesFrom <http://www.schiphol.nl/ontology/ontology#Locatie>
                  ] ;
      rdfs:label "Eis"@nl ,
              "Requirement"@en .

###  http://www.schiphol.nl/ontology#Elevator
aas:Elevator rdf:type owl:Class ;
          rdfs:subClassOf ifc:IfcTransportElement ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty aas:IsContainedBy ;
                        owl:someValuesFrom <https://w3id.org/bot#Zone>
                      ] ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty aas:contains ;
                        owl:someValuesFrom ifc:IfcDoor
                      ] ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty aas:requires ;
                        owl:someValuesFrom aas:LegalObligations
                      ] ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty aas:requires ;
                        owl:someValuesFrom aas:Warranty
                      ] .

###  http://www.schiphol.nl/ontology#Existing
aas:Existing rdf:type owl:Class ;
          rdfs:subClassOf aas:Status .

###  http://www.schiphol.nl/ontology#FireRating
aas:FireRating rdf:type owl:Class ;
           rdfs:subClassOf aas:Kenmerk ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty aas:hasValue ;
                         owl:allValuesFrom xsd:int
                       ] .

###  http://www.schiphol.nl/ontology#Function
aas:Function rdf:type owl:Class ;
          rdfs:subClassOf aas:InformationItem .

###  http://www.schiphol.nl/ontology#InformationItem
aas:InformationItem rdf:type owl:Class .

###  http://www.schiphol.nl/ontology#IsExternal
aas:IsExternal rdf:type owl:Class ;
           rdfs:subClassOf aas:Kenmerk ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty aas:hasBoolean ;
                         owl:allValuesFrom xsd:boolean
                       ] .

###  http://www.schiphol.nl/ontology#Kenmerk
aas:Kenmerk rdf:type owl:Class ;
         rdfs:label "Characteristic"@en ,
                "Kenmerk"@nl .
```

```
### http://www.schiphol.nl/ontology#LegalObligations
aas:LegalObligations rdf:type owl:Class ;
            rdfs:subClassOf aas:Eis .

### http://www.schiphol.nl/ontology#LoadBearing
aas:LoadBearing rdf:type owl:Class ;
         rdfs:subClassOf aas:Kenmerk ,
                 [ rdf:type owl:Restriction ;
                   owl:onProperty aas:hasValue ;
                   owl:allValuesFrom xsd:boolean
                 ] .

### http://www.schiphol.nl/ontology#Measurement
aas:Measurement rdf:type owl:Class ;
         rdfs:subClassOf aas:Kenmerk ,
                 [ rdf:type owl:Restriction ;
                   owl:onProperty aas:hasValue ;
                   owl:allValuesFrom xsd:decimal
                 ] .

### http://www.schiphol.nl/ontology#Movingwalkway
aas:Movingwalkway rdf:type owl:Class ;
          rdfs:subClassOf ifc:IfcTransportElement .
### http://www.schiphol.nl/ontology#New
aas:New rdf:type owl:Class ;
    rdfs:subClassOf aas:Status .

### http://www.schiphol.nl/ontology#Project
aas:Project rdf:type owl:Class ;
       rdfs:subClassOf aas:InformationItem ,
               [ rdf:type owl:Restriction ;
                 owl:onProperty aas:hasString ;
                 owl:someValuesFrom xsd:string
               ] .

### http://www.schiphol.nl/ontology#STD
aas:STD rdf:type owl:Class ;
    rdfs:subClassOf aas:Kenmerk ;
    rdfs:comment "Schiphol Technical Decomposition, which is a variance on the NL-SfB classification system."@
en .

### http://www.schiphol.nl/ontology#Site
aas:Site rdf:type owl:Class ;
     rdfs:subClassOf <https://w3id.org/bot#Zone> ;
     owl:disjointWith aas:Space ,
             aas:Storey .

### http://www.schiphol.nl/ontology#Space
aas:Space rdf:type owl:Class ;
     rdfs:subClassOf <https://w3id.org/bot#Zone> ;
     owl:disjointWith aas:Storey .

### http://www.schiphol.nl/ontology#Status
aas:Status rdf:type owl:Class ;
      rdfs:subClassOf aas:InformationItem ,
              [ rdf:type owl:Restriction ;
                owl:onProperty aas:hasString ;
                owl:someValuesFrom xsd:string
```

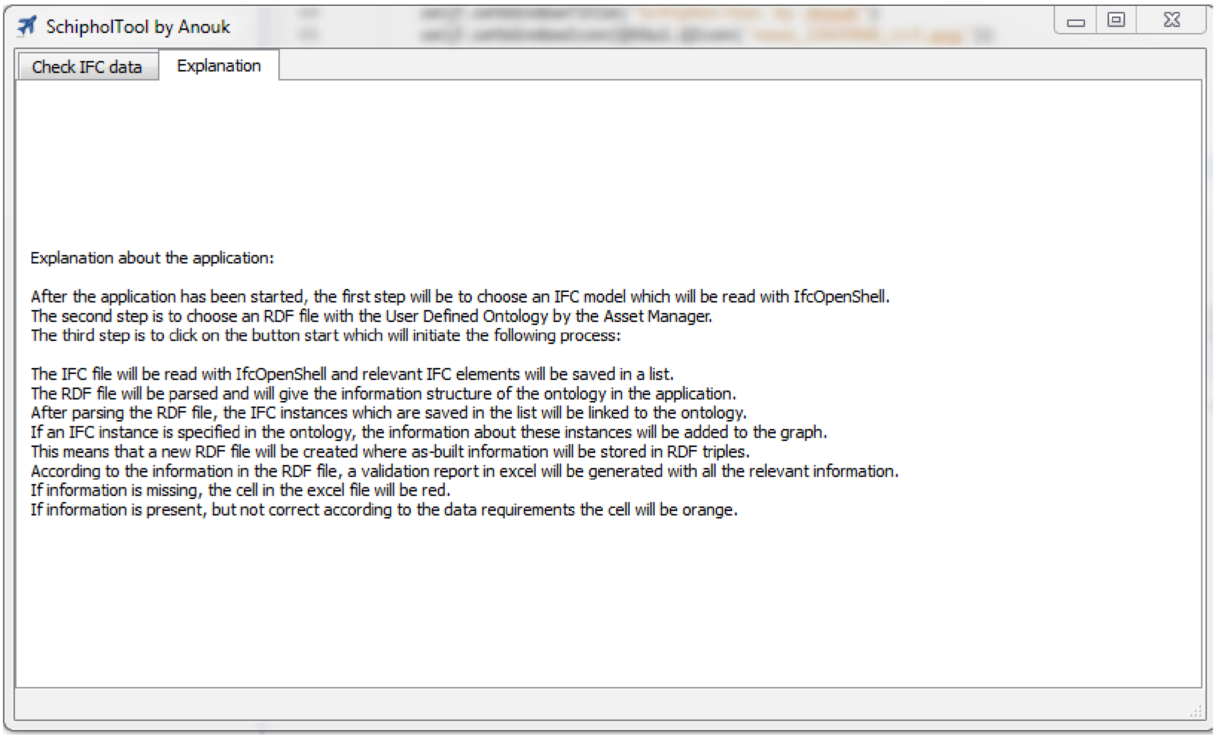
```
                    ] .

###  http://www.schiphol.nl/ontology#Storey
aas:Storey rdf:type owl:Class ;
        rdfs:subClassOf <https://w3id.org/bot#Zone> .

###  http://www.schiphol.nl/ontology#Temporary
aas:Temporary rdf:type owl:Class ;
          rdfs:subClassOf aas:Status .

###  http://www.schiphol.nl/ontology#Warranty
aas:Warranty rdf:type owl:Class ;
          rdfs:subClassOf aas:Eis .

###  http://www.schiphol.nl/ontology#inUse
aas:inUse rdf:type owl:Class ;
       rdfs:subClassOf aas:Status .

###  http://www.schiphol.nl/ontology/ontology#Locatie
<http://www.schiphol.nl/ontology/ontology#Locatie> rdf:type owl:Class ;
                                     rdfs:subClassOf owl:Thing ;
                                     rdfs:label "Locatie"@nl ,
                                            "Location"@en .

###  https://w3id.org/bot#Zone
<https://w3id.org/bot#Zone> rdf:type owl:Class ;
                    rdfs:subClassOf <http://www.schiphol.nl/ontology/ontology#Locatie> .

#################################################################
#    Individuals
#################################################################

###  http://www.schiphol.nl/ontology#Elevator
aas:Elevator rdf:type owl:NamedIndividual ,
                aas:Elevator .

###  http://www.schiphol.nl/ontology#ElevatorShaft
aas:ElevatorShaft rdf:type owl:NamedIndividual .

###  http://www.schiphol.nl/ontology#Escalator
aas:Escalator rdf:type owl:NamedIndividual .

###  http://www.schiphol.nl/ontology#FireCompartment
aas:FireCompartment rdf:type owl:NamedIndividual ,
                    <https://w3id.org/bot#Zone> .

###  http://www.schiphol.nl/ontology#Movingwalkway
aas:Movingwalkway rdf:type owl:NamedIndividual ,
                   aas:Movingwalkway .

###  Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi
```

Appendix IV: Graphical User Interface (GUI) of the application

The first tab;



The second tab;



Appendix V: Python Script

```
import sys
import uuid
import csv
import datetime

from PyQt4 import QtGui

from openpyxl import Workbook
from openpyxl.styles import PatternFill, GradientFill, colors, Font, Color, Fill
from openpyxl.cell import Cell
from openpyxl.drawing.image import Image

from rdflib import Graph, Literal, URIRef, Namespace, XSD, RDF
from rdflib.namespace import NamespaceManager
from rdflib.term import URIRef

import ifcopenshell
import ifcopenshell.geom

settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)

#namespace of the ontology
aas = Namespace("http://www.schiphol.nl/ontology#")

#list with asked ifcelements
BElem = []
wrong_datatype = []

#--------------------Initiating the application --------------------------------

#Main class of the application
class initUI(object):
    def __init__(self, *args):
        # Constructing an application
        app = QtGui.QApplication(sys.argv)

        # Viewer initialization
        self.main = Main(self)
        self.main.show()
        self.main.statusBar()

        #Raises a system exit
        sys.exit(app.exec_())

#Main Class of the Graphical User Interface
class Main(QtGui.QMainWindow):
    def __init__(self, parent=None):
        self.parent=parent

        #Instantiating the tabs
        global filename
        self.filename = None

        self.tabs = QtGui.QTabWidget()
```

```
        self.tab_ifc = QtGui.QWidget()
        self.tab_explanation = QtGui.QWidget()

        self.tabs.addTab(self.tab_ifc, 'Check IFC data')
        self.tabs.addTab(self.tab_explanation, 'Explanation')

        #Viewer initialization
        super(Main, self).__init__()
        self.setGeometry(200, 300, 800, 450)
        self.setWindowTitle("SchipholTool by Anouk")
         self.setWindowIcon(QtGui.QIcon('noun_1565960_cc3.png'))
        #self.setStyleSheet("background-color: White;")

        #Calling the tab methods
        #first tab
        self.ifc_tab()
        #second tab
        self.explanation_tab()

        self.setCentralWidget(self.tabs)

#----------------------------Tab_1-----------------------------------------------
    def ifc_tab(self):
        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()
        h2box = QtGui.QHBoxLayout()
        h3box = QtGui.QHBoxLayout()
        h4box = QtGui.QHBoxLayout()

        vbox.addLayout(hbox)
        vbox.addLayout(h2box)
        vbox.addLayout(h3box)
        vbox.addLayout(h4box)

        self.tab_ifc.setLayout(vbox)

        # textual introduction of the tab and functionalities
        ifc_tab_introduction = QtGui.QLabel( '''This tool will check the IFC file
loaded by the user against the Schiphol User Defined Ontology
and will generate a validation report which will state the information in the IFC
file, the information which is incorrect and which information is missing'''
        )
        ifc_tab_introduction.setWordWrap(True)
        ifc_tab_introduction.setMaximumHeight(100)

        open_ifc_btn_title = QtGui.QLabel("Open an IFC file which has to be
checked:")
        open_ifc_btn = QtGui.QPushButton("Open IFC", self)
        open_ifc_btn.clicked.connect(self.open_ifc_file)

        open_rdf_btn_title = QtGui.QLabel("Open an RDF file which includes the
UDO:")
        open_rdf_btn = QtGui.QPushButton("Open RDF", self)
        open_rdf_btn.clicked.connect(self.open_rdf_file)
        #open_rdf_btn.setStyleSheet("background-color: blue")

        Write_btn_title = QtGui.QLabel("After opening an IFC and RDF file choose
Start:")
```

```
        Write_btn = QtGui.QPushButton("Start", self)
        Write_btn.clicked.connect(self.create_RDF_information)

        self.show()

        hbox.addWidget(ifc_tab_introduction)
        h2box.addWidget(open_ifc_btn_title)
        h2box.addWidget(open_ifc_btn)
        h3box.addWidget(open_rdf_btn_title)
        h3box.addWidget(open_rdf_btn)
        h4box.addWidget(Write_btn_title)
        h4box.addWidget(Write_btn)

    # Loading an IFC file in the viewer application
    def open_ifc_file(self, filename=None):
        self.filename = QtGui.QFileDialog.getOpenFileName(self, 'Open file', ".",
"Industry Foundation Classes (*.ifc)")
        self.reader_ifc(self.filename)

    # Inititate RDF file based on UDO
    def open_rdf_file(self, filename=None):
        self.filenameRDF = QtGui.QFileDialog.getOpenFileName(self, 'Open file', ".",
"Resource Description Framework(*.ttl)")
        g = Graph()
        g.parse(str(self.filenameRDF), format="turtle")
        g.namespace_manager.bind('aas', aas)

        if not self.filename: QtGui.QMessageBox.warning(self,
                                                       "No IFC file loaded!",
                                                       "Load a model first!")

        return

    # Reader of the ifc file
    def reader_ifc(self, filename):
        self.ifc_file = ifcopenshell.open(filename)
        #elements = self.ifc_file.by_type('IfcProduct')
        for B_elem in self.ifc_file.by_type("IfcBuildingElement") :
            if B_elem.is_a("IfcDoor"):
                BElem.append(B_elem)
            if B_elem.is_a("IfcWall"):
                if B_elem.is_a ("IfcWallStandardCase"):
                    pass
                else:
                    BElem.append(B_elem)
            if B_elem.is_a("IfcCurtainWall"):
                BElem.append(B_elem)
        for T_elem in self.ifc_file.by_type("IfcTransportElement") :
            BElem.append(T_elem)

    #Initiate RDF file to save IFC instances in RDF-----------------------------
    def create_RDF_information(self):
        g = Graph()

        #Inititate rdf file
        rdffile = open("ifcotl.ttl", "w")

        g = Graph()
        g.parse(str(self.filenameRDF), format="ttl")
```

```
        g.namespace_manager.bind('aas', aas)

        #Identify URIRefs
        elevator2 = URIRef("http://www.schiphol.nl/ontology#Elevator")
        door = URIRef("http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcDoor")
        wall = URIRef("http://ifcowl.openbimstandards.org/IFC4_ADD2#IfcWall")
        buildingElement = URIRef("http://ifcowl.openbimstandards.org/IFC4_
ADD2#IfcBuildingElement")
        movingwalkway = URIRef("http://www.schiphol.nl/ontology#Movingwalkway")

        number = 1

        #Add Triples in g based on selected ifc instances of IfcBuildingElement
        for listb in BElem:
            listB=listb.get_info()

            #generate URI's
            if number < 10:
                bEl = URIRef(aas + (listB['type']) + "00" + str(number))
            elif number < 100:
                bEl = URIRef(aas + (listB['type']) + "0" + str(number))
            else:
                bEl = URIRef(aas + (listB['type']) + str(number))

            #add correct entity instance
            if listB['type'] == 'IfcDoor':
                g.add((bEl, RDF.type, door))
                g.add((bEl, aas.ClearHeight, Literal(listB['OverallHeight'],
datatype=XSD.float)))
                g.add((bEl, aas.ClearWidth, Literal(listB['OverallWidth'],
datatype=XSD.float)))
            if listB['type'] == 'IfcWall':
                g.add((bEl, RDF.type, wall))
            if listB['type'] == "IfcTransportElement":
                if listB["OperationType"] == 'ELEVATOR':
                    g.add((bEl, RDF.type, elevator2))
                    g.add((bEl, aas.capacitynr, Literal(listB['CapacityByNumber'],
datatype=XSD.float)))
                    g.add((bEl, aas.capacitywght,
Literal(listB['CapacityByWeight'], datatype=XSD.float)))
                if listB["OperationType"] == 'MOVINGWALKWAY':
                    g.add((bEl, RDF.type, movingwalkway))

            g.add((bEl, aas.hasGuid, Literal(listB['GlobalId'], datatype=XSD.
unicode)))
            g.add((bEl, aas.hasName, Literal(listB['Name'], datatype=XSD.string)))
            g.add((bEl, aas.hasType, Literal(listB['ObjectType'], datatype=XSD.
string)))
            g.add((bEl, aas.requires, Literal(listB['Description'], datatype=XSD.
string)))

            number+=1

            #add storey& SPACE
            for RelContainedInSpatialStructure in listb.ContainedInStructure:
                if RelContainedInSpatialStructure.RelatingStructure.
is_a("IfcBuildingStorey"):
                    g.add((bEl, aas.hasLocation,
```

```
Literal(RelContainedInSpatialStructure.RelatingStructure['Name'], datatype=XSD.
string)))
                if RelContainedInSpatialStructure.RelatingStructure.
is_a("IfcSpace"):
                    g.add((bEl, aas.inSpace,
Literal(RelContainedInSpatialStructure.RelatingStructure['Name'], datatype=XSD.
string)))

            #add building
            for storey in self.ifc_file.by_type("IfcBuildingStorey"):
                for RelAggregates in storey.Decomposes:
                    g.add((bEl, aas.hasBuilding, Literal(RelAggregates.
RelatingObject['LongName'], datatype=XSD.string)))

            #add project
            for project in self.ifc_file.by_type("IfcProject"):
                g.add((bEl, aas.project, Literal(project.LongName, datatype=XSD.
string)))

            #add classification
            for rel in listb.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    if rel.RelatingPropertyDefinition.
is_a("IfcElementQuantity"):continue
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.Name == "Assembly Code":
                            g.add((bEl, aas.hasSTD, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.float)))
                         elif prop.Name == "nl-sfb code":
                    g.add((bEl, aas.hasSTD, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.number)))
                        else:
                            g.add((bEl, aas.hasSTD, Literal(" ", datatype=XSD.
float)))
                        if prop.Name == "IsExternal":
                            g.add((bEl, aas.isExternal, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.boolean)))
                        if prop.Name == "LoadBearing":
                            g.add((bEl, aas.loadBearing, Literal(prop.
NominalValue.wrappedValue, datatype=XSD.boolean)))
                        if prop.Name == "FireRating":
                            g.add((bEl, aas.fireRating, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.string)))
                        if prop.Name == "AcousticRating":
                            g.add((bEl, aas.fireRating, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.string)))

                        #add quantities
                        if prop.Name == "Height":
                            g.add((bEl, aas.ClearHeight, Literal(prop.
NominalValue.wrappedValue, datatype=XSD.float)))
                        if prop.Name == "Width":
                            g.add((bEl, aas.ClearWidth, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.float)))
                        if prop.Name == "Depth":
                            g.add((bEl, aas.ClearDepth, Literal(prop.NominalValue.
wrappedValue, datatype=XSD.float)))
```

```
        # End of the creation of the rdf file!!!
        a = g.serialize(format='turtle')
        rdffile.write(a)
        rdffile.close()

        #query all the basis requirements
        with open(r'C:\Users\s122019\QueryResult_Basic.csv', 'w') as testfile1:
            headers = ['bEL', 'GlobalId', 'Name', 'ObjectType', 'STD', 'Project' ,
'Building', 'Storey', 'Space', 'ClearWidth', 'ClearHeight', 'ClearDepth']
            csv_writer = csv.DictWriter(testfile1, delimiter=';',
fieldnames=headers)

            for row in g.query("""
                PREFIX aas: <http://www.schiphol.nl/ontology#>
                PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
                PREFIX owl: <http://www.w3.org/2002/07/owl#>
                PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
                PREFIX xml: <http://www.w3.org/XML/1998/namespace>
                PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
                PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>


                SELECT DISTINCT ?bEL ?GlobalId ?Name ?ObjectType ?STD ?Project
?Building ?Storey ?Space ?ClearWidth ?ClearHeight ?ClearDepth
                WHERE {
                    ?bEL aas:hasGuid ?GlobalId .
                    ?bEL aas:hasName ?Name .
                    ?bEL aas:hasType ?ObjectType .
                    ?bEL aas:hasSTD ?STD .
                    ?bEL aas:project ?Project .
                    ?bEL aas:hasBuilding ?Building.
                    ?bEL aas:hasLocation ?Storey .
                    OPTIONAL {?bEL aas:inSpace ?Space . }
                    OPTIONAL {?bEL aas:ClearWidth ?ClearWidth . }
                    OPTIONAL {?bEL aas:ClearHeight ?ClearHeight . }
                    OPTIONAL {?bEL aas:ClearDepth ?ClearDepth . }
                    } ORDER BY ?bEL
            """):
                csv_writer.writerow({'bEL':row[0], 'GlobalId':row[1],
'Name':row[2], 'ObjectType':row[3], 'STD':row[4], 'Project':row[5],
'Building':row[6], 'Storey':row[7], 'Space':row[8], 'ClearWidth':row[9],
'ClearHeight':row[10], 'ClearDepth':row[11]})


    #-------------------------Generate Excel report from csv data-----------------
        wb = Workbook()
        ws = wb.active
        ws.title = "Results of SchipholTool"

        ft1 = Font(name='Arial', size=16, bold=True)
        ws.merge_cells('A1:L1')
        ws.row_dimensions[1].height = 60
        ws.row_dimensions[2].height = 25
        #Put in a logo
        img = Image('Schiphol-AA-Logo-RGB.jpg')
        img.width = 200
        img.height = 75
        ws.add_image(img, 'A1')
```

```
#Title
ws.merge_cells('A2:L2')
ws['A2'] = 'Validation Report '
a1 = ws['A2']
a1.font = ft1

#datetime
dateandtime = datetime.datetime.now().strftime("%Y-%m-%d, %H:%M")
ws['A3'] = dateandtime
ft_bold = Font(bold=True)
a3 = ws['A3']
a3.font = ft_bold

#Headers of cells
ws['A4'] = 'OBJECT'
ws['B4'] = 'GLOBALID'
ws['C4'] = 'NAME'
ws['D4'] = 'OBJECTTYPE'
ws['E4'] = 'STD'
ws['F4'] = 'PROJECT'
ws['G4'] = 'BUILDING'
ws['H4'] = "STOREY"
ws['I4'] = "SPACE"
ws['J4'] = 'CLEARWIDTH'
ws['K4'] = 'CLEARHEIGHT'
ws['L4'] = 'CLEARDEPTH'

#make headers bold
for name in ws["4:4"]:
    name.font = ft_bold

fill_Darkred = PatternFill("solid", fgColor='00800000')
fill_Orange = PatternFill("solid", fgColor='00FF9900')

with open(r'C:\Users\s122019\QueryResult_Basic.csv', 'r') as f:
    for rows in csv.reader(f, delimiter=';'):
        ws.append(rows)
        for row in ws.iter_rows(min_row=1, max_col=12):
            for cell in row:
                if cell._value == "None":
                    cell.fill = fill_Darkred
                elif cell._value == "":
                    cell.fill = fill_Darkred
                elif cell._value == " ":
                    cell.fill = fill_Darkred
                elif cell._value == "[]":
                    cell.fill = fill_Darkred
                elif cell._value == "0.0":
                    cell.fill = fill_Orange

        #check if globalId is correct in row B
        for column in ws.iter_cols(min_col=2, max_col=2):
            for cell in column:
                if type(cell) is None:
                    pass
            if type(cell._value) != unicode:
                cell.fill = fill_Orange
```

```
                elif len(cell._value) != 22:
                    #print “no”, cell
                    cell.fill = fill_Orange

            #check if Name & Type is correct in row C & D
            for column in ws.iter_cols(min_col=3, max_col=4):
                for cell in column:
                    if type(cell) is None:
                        pass
                if type(cell._value) != unicode:
                    cell.fill = fill_Orange

            #check if STD is correct in row E
            for column in ws.iter_cols(min_col=5, max_col=5):
                for cell in column:
                    if type(cell) is None:
                        pass
                with open(r’list-STD-codes.csv’, ‘rb’) as checkfile:
                    # using the first row as keys (“Class-codenotatie”)
                    reader = csv.DictReader(checkfile,delimiter=”;”)
                    # iterate over all rows
                    for r in reader:
                        if cell._value != r [“Class-codenotatie”]:
                            cell.fill = fill_Orange

            #check if Project & Building & Storey & Space is correct in row F
& G & H & I
            for column in ws.iter_cols(min_col=6, max_col=9):
                for cell in column:
                    if type(cell) is None:
                        pass
                if type(cell._value) != unicode:
                    cell.fill = fill_Orange

            #check if Width, Height & Depth is correct in row J & K & L
            #should be any number (float)
            for column in ws.iter_cols(min_col=10, max_col=12):
                for cell in column:
                    if type(cell) is None:
                        pass
                if type(cell._value) != unicode:
                    cell.fill = fill_Orange

    #Dimensions of cells
    ws.column_dimensions[“A”].width = 60.0
    ws.column_dimensions[“B”].width = 30.0
    ws.column_dimensions[“C”].width = 100.0
    ws.column_dimensions[“D”].width = 60.0
    ws.column_dimensions[“E”].width = 20.0
    ws.column_dimensions[“F”].width = 30.0
    ws.column_dimensions[“G”].width = 25.0
    ws.column_dimensions[“H”].width = 25.0
    ws.column_dimensions[“I”].width = 25.0
    ws.column_dimensions[“J”].width = 25.0
    ws.column_dimensions[“K”].width = 25.0
    ws.column_dimensions[“L”].width = 25.0

    wb.save(‘QueryResult.xlsx’)
```

```
#---------------------------Tab_2---------------------------------------------
    def explanation_tab(self):
        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()

        vbox.addLayout(hbox)

        self.tab_explanation.setLayout(vbox)

        # textual introduction of the tab and functionalities
        explanation_tab_introduction = QtGui.QLabel( ‘’’Explanation about the
application:

After the application has been started, the first step will be to choose an IFC
model which will be read with IfcOpenShell.
The second step is to choose an RDF file with the User Defined Ontology by the Asset
Manager.
The third step is to click on the button start which will initiate the following
process:

The IFC file will be read with IfcOpenShell and relevant IFC elements will be saved
in a list.
The RDF file will be parsed and will give the information structure of the ontology
in the application.
After parsing the RDF file, the IFC instances which are saved in the list will be
linked to the ontology.
If an IFC instance is specified in the ontology, the information about these
instances will be added to the graph.
This means that a new RDF file will be created where as-built information will be
stored in RDF triples.
According to the information in the RDF file, a validation report in excel will be
generated with all the relevant information.
If information is missing, the cell in the excel file will be red.
If information is present, but not correct according to the data requirements the
cell will be orange. ‘’’
        )
        explanation_tab_introduction.setWordWrap(True)
        explanation_tab_introduction.setMaximumHeight(200)

        self.show()

        hbox.addWidget(explanation_tab_introduction)

#---------------------------End Tab_2-------------------------------------------

init = initUI()


if __name__ == ‘__Main__’:
    Main()

    print(“------GENERATED-------”)
    #sys.exit(app.exec_())
```

Appendix VI: Query of the python script

```
#query all the basis requirements
with open(r'C:\Users\s122019\QueryResult_Basic.csv', 'w') as testfile1:
    headers = ['bEl', 'GlobalId', 'Name', 'ObjectType', 'STD', 'Project'
    ,
'Building', 'Storey', 'Space', 'ClearWidth', 'ClearHeight', 'ClearDepth']  csv_
    writer = csv.DictWriter(testfile1, delimiter=';', fieldnames=headers)

    for row in g.query("""
PREFIX aas: <http://www.schiphol.nl/ontology#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>  PREFIX owl: <http://
www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX xml: <http://www.
w3.org/XML/1998/namespace>   PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>


        SELECT DISTINCT ?bEl ?GlobalId ?Name ?ObjectType ?STD ?Project ?Building
 ?Storey ?Space ?ClearWidth ?ClearHeight ?ClearDepth
        WHERE {
            ?bEl aas:hasGuid ?GlobalId .
            ?bEl aas:hasName ?Name .
            ?bEl aas:hasType ?ObjectType .
            ?bEl aas:hasSTD ?STD .
            ?bEl aas:project ?Project .
            ?bEl aas:hasBuilding ?Building.
            ?bEl aas:hasLocation ?Storey .
            OPTIONAL { ?bEl aas:inSpace ?Space
            . }
            OPTIONAL {?bEl aas:ClearWidth ?ClearWidth . }
            OPTIONAL {?bEl aas:ClearHeight ?ClearHeight . }
            OPTIONAL {?bEl aas:ClearDepth ?ClearDepth . }
            } ORDER BY ?bEl
    """):
        csv_writer.writerow({'bEl':row[0],      'GlobalId':row[1],
        'Name':row[2],
 'ObjectType':row[3],      'STD':row[4],      'Project':row[5],
 'Building':row[6],
 'Storey':row[7],    'Space':row[8],    'ClearWidth':row[9],
 'ClearHeight':row[10],  'ClearDepth':row[11]})
```

Appendix VII: Partial RDF files

Partial file of the elevator model of the KLM-ICA lounge:

```
@prefix : <http://www.schiphol.nl/ontology#> .
@prefix aas: <http://www.schiphol.nl/ontology#> .
@prefix ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

aas:IfcTransportElement001 a aas:Elevator ;
  aas:capacitynr “0.0”^^xsd:float ;
  aas:capacitywght “0.0”^^xsd:float ;
  aas:hasBuilding “”^^xsd:string ;
  aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_M”^^xsd:unicode ;
  aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
  aas:hasName “66_KON_MOD_MACHINE_MX14_MONO:66_KON_MOD_MACHINE_MX14_
MONO:416739”^^xsd:string ;
  aas:hasSTD “”^^xsd:number ;
  aas:hasType “66_KON_MOD_MACHINE_MX14_MONO”^^xsd:string ;
  aas:project “ICA Lounge Schiphol”^^xsd:string ;
  aas:requires “None”^^xsd:string .

aas:IfcTransportElement002 a aas:Elevator ;
  aas:ClearHeight “2300.0”^^xsd:float ;
  aas:capacitynr “0.0”^^xsd:float ;
  aas:capacitywght “0.0”^^xsd:float ;
  aas:hasBuilding “”^^xsd:string ;
  aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_T”^^xsd:unicode ;
  aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
  aas:hasName “66_KON_MOD_CWT:66_KON_MOD_CWT:416744”^^xsd:string ;
  aas:hasSTD “”^^xsd:number ;
  aas:hasType “66_KON_MOD_CWT”^^xsd:string ;
  aas:project “ICA Lounge Schiphol”^^xsd:string ;
  aas:requires “None”^^xsd:string .

aas:IfcTransportElement003 a aas:Elevator ;
  aas:capacitynr “0.0”^^xsd:float ;
  aas:capacitywght “0.0”^^xsd:float ;
  aas:hasBuilding “”^^xsd:string ;
  aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_V”^^xsd:unicode ;
  aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
  aas:hasName “66_KON_MOD_Buffer_Extension_KM812340G06:66_KON_MOD_Buffer_Extension_
KM812340G06:416746”^^xsd:string ;
  aas:hasSTD “”^^xsd:number ;
  aas:hasType “66_KON_MOD_Buffer_Extension_KM812340G06”^^xsd:string ;
  aas:project “ICA Lounge Schiphol”^^xsd:string ;
  aas:requires “None”^^xsd:string .

aas:IfcTransportElement004 a aas:Elevator ;
  aas:capacitynr “0.0”^^xsd:float ;
  aas:capacitywght “0.0”^^xsd:float ;
  aas:hasBuilding “”^^xsd:string ;
  aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_P”^^xsd:unicode ;
  aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
  aas:hasName “66_KON_MOD_Shaft_Light:66_KON_MOD_Shaft_Light:416748”^^xsd:string ;
```

```
    aas:hasSTD “”^^xsd:number ;
    aas:hasType “66_KON_MOD_Shaft_Light”^^xsd:string ;
    aas:project “ICA Lounge Schiphol”^^xsd:string ;
    aas:requires “None”^^xsd:string .

aas:IfcTransportElement005 a aas:Elevator ;
    aas:capacitynr “0.0”^^xsd:float ;
    aas:capacitywght “0.0”^^xsd:float ;
    aas:hasBuilding “”^^xsd:string ;
    aas:hasGuid “1ClOvoqvXDFfhj96CrwQ_4”^^xsd:unicode ;
    aas:hasLocation “01 eerste verdieping EF”^^xsd:string ;
                aas:hasName     “66_KON_MOD_Buffer_Extension_KM812340G06:66_KON_MOD_Buffer_Extension_
KM812340G06:416753”^^xsd:string ;
    aas:hasSTD “”^^xsd:number ;
    aas:hasType “66_KON_MOD_Buffer_Extension_KM812340G06”^^xsd:string ;
    aas:project “ICA Lounge Schiphol”^^xsd:string ;
    aas:requires “None”^^xsd:string .
………………………………
aas:IfcTransportElement080 a aas:Elevator ;
    aas:capacitynr “0.0”^^xsd:float ;
    aas:capacitywght “0.0”^^xsd:float ;
    aas:hasBuilding “”^^xsd:string ;
    aas:hasGuid “0YW_WONB95hwPeyhymLw9r”^^xsd:unicode ;
    aas:hasLocation “03 derde verdieping EF”^^xsd:string ;
    aas:hasName “66_KON_MOD_Car_Door_2L:66_KON_MOD_Car_Door_2L:3495993”^^xsd:string ;
    aas:hasSTD “”^^xsd:number ;
    aas:hasType “66_KON_MOD_Car_Door_2L”^^xsd:string ;
    aas:project “ICA Lounge Schiphol”^^xsd:string ;
    aas:requires “None”^^xsd:string .

aas:IfcTransportElement081 a aas:Elevator ;
    aas:capacitynr “0.0”^^xsd:float ;
    aas:capacitywght “0.0”^^xsd:float ;
    aas:hasBuilding “”^^xsd:string ;
    aas:hasGuid “0YW_WONB95hwPeyhymLw9t”^^xsd:unicode ;
    aas:hasLocation “03 derde verdieping EF”^^xsd:string ;
                    aas:hasName     “66_KON_MOD_LDO_KES800_2R_P31_N3:66_KON_MOD_LDO_KES800_2R_P31_
N3:3495995”^^xsd:string ;
    aas:hasSTD “”^^xsd:number ;
    aas:hasType “66_KON_MOD_LDO_KES800_2R_P31_N3”^^xsd:string ;
    aas:project “ICA Lounge Schiphol”^^xsd:string ;
    aas:requires “None”^^xsd:string .
```

Partial file of the elevator model of P3:

```
@prefix : <http://www.schiphol.nl/ontology#> .
@prefix aas: <http://www.schiphol.nl/ontology#> .
@prefix ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

aas:IfcDoor001 a ifc:IfcDoor ;
    aas:ClearHeight “2410.0”^^xsd:float ;
    aas:ClearWidth “1320.0”^^xsd:float ;
    aas:hasBuilding “Parkeergarage P3”^^xsd:string ;
```

aas:hasGuid “0CQpf9SY1DgOU1hBpiJV”^^xsd:unicode ;
aas:hasLocation “00 begane grond”^^xsd:string ;
aas:hasName “66_KON_MOD_Rechte muurkap (bilnaad):66_KON_MOD_Muurkap
(Zincorplaat):401224”^^xsd:string ;
aas:hasSTD “ “^^xsd:number ;
aas:hasType “66_KON_MOD_Muurkap (Zincorplaat)”^^xsd:string ;
aas:isExternal false ;
aas:project “Schiphol parkeergarage P3”^^xsd:string ;
aas:requires “None”^^xsd:string .

aas:IfcDoor002 a ifc:IfcDoor ;
aas:ClearHeight “2410.0”^^xsd:float ;
aas:ClearWidth “1320.0”^^xsd:float ;
aas:hasBuilding “Parkeergarage P3”^^xsd:string ;
aas:hasGuid “0CQpf9SY1DgOU1hBpiJU”^^xsd:unicode ;
aas:hasLocation “00 begane grond”^^xsd:string ;
aas:hasName “66_KON_MOD_Rechte muurkap (bilnaad):66_KON_MOD_Muurkap
(Zincorplaat):401225”^^xsd:string ;
aas:hasSTD “ “^^xsd:number ;
aas:hasType “66_KON_MOD_Muurkap (Zincorplaat)”^^xsd:string ;
aas:isExternal false ;
aas:project “Schiphol parkeergarage P3”^^xsd:string ;
aas:requires “None”^^xsd:string .
……………………
aas:IfcTransportElement015 a aas:Elevator ;
aas:capacitynr “0.0”^^xsd:float ;
aas:capacitywght “0.0”^^xsd:float ;
aas:hasBuilding “Parkeergarage P3”^^xsd:string ;
aas:hasGuid “0CQpf9SY1DgOU1hBpiIM”^^xsd:unicode ;
aas:hasLocation “00 begane grond”^^xsd:string ;
aas:hasName “66_KON_MOD_42306675-lift 2:66_KON_MOD_42306675-lift 2:401153”^^xsd:string ;
aas:hasSTD “ “^^xsd:number ;
aas:hasType “66_KON_MOD_42306675-lift 2”^^xsd:string ;
aas:project “Schiphol parkeergarage P3”^^xsd:string ;
aas:requires “None”^^xsd:string .

aas:IfcTransportElement016 a aas:Elevator ;
aas:ClearDepth “62.0”^^xsd:float ;
aas:ClearWidth “82.6666666667”^^xsd:float ;
aas:capacitynr “0.0”^^xsd:float ;
aas:capacitywght “0.0”^^xsd:float ;
aas:hasBuilding “Parkeergarage P3”^^xsd:string ;
aas:hasGuid “0CQpf9SY1DgOU1hBpiIL”^^xsd:unicode ;
aas:hasLocation “00 begane grond”^^xsd:string ;
aas:hasName “66_KON_MOD_CWT_guiderail:66_KON_MOD_CWT_guiderail:401154”^^xsd:string ;
aas:hasSTD “ “^^xsd:number ;
aas:hasType “66_KON_MOD_CWT_guiderail”^^xsd:string ;
aas:project “Schiphol parkeergarage P3”^^xsd:string ;
aas:requires “None”^^xsd:string .
……………………..

Partial file of the architectural model of P3:

@prefix : <http://www.schiphol.nl/ontology#> .
@prefix aas: <http://www.schiphol.nl/ontology#> .
@prefix ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

aas:IfcCurtainWall027
   aas:fireRating “30”^^xsd:string ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “18nj49GpTEqxkL6Njb59e0”^^xsd:unicode ;
   aas:hasLocation “00 begane grond”^^xsd:string ;
   aas:hasName “Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:7825401”^^xsd:string ;
   aas:hasType “Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8215902”^^xsd:string ;
   aas:isExternal true ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .

aas:IfcCurtainWall030
   aas:fireRating “60”^^xsd:string ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “18nj49GpTEqxkL6Njb59IT”^^xsd:unicode ;
   aas:hasLocation “00 begane grond”^^xsd:string ;
   aas:hasName “Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7825764”^^xsd:string ;
   aas:hasType “Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822”^^xsd:string ;
   aas:isExternal true ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .
……………………..

as:IfcDoor011 a ifc:IfcDoor ;
   aas:ClearHeight “2350.0”^^xsd:float ;
   aas:ClearWidth “2200.0”^^xsd:float ;
   aas:fireRating “60”^^xsd:string ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “0ZLDNXVMD4$geq9hXI8L7c”^^xsd:unicode ;
   aas:hasLocation “01 eerste verdieping”^^xsd:string ;
   aas:hasName “BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7703961”^^xsd:string ;
   aas:hasSTD “31.32 “^^xsd:number ;
   aas:hasType “BN_OPG_Schuifdeur”^^xsd:string ;
   aas:isExternal true ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .

aas:IfcDoor012 a ifc:IfcDoor ;
   aas:ClearHeight “2385.0”^^xsd:float ;
   aas:ClearWidth “2506.0”^^xsd:float ;
   aas:fireRating “60”^^xsd:string ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “0ZLDNXVMD4$geq9hXI8L7d”^^xsd:unicode ;
   aas:hasLocation “01 eerste verdieping”^^xsd:string ;
   aas:hasName “BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7703962”^^xsd:string ;
   aas:hasSTD “31.31”^^xsd:number ;
   aas:hasType “BN_Deur met zijlicht 01”^^xsd:string ;
   aas:isExternal true ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .
……………………..

aas:IfcWall001 a ifc:IfcWall ;
   aas:hasBuilding “”^^xsd:string ;
```

```
   aas:hasGuid “1MlDW4hyz7thdXm8zl0rs_”^^xsd:unicode ;
   aas:hasLocation “02 tweede verdieping”^^xsd:string ;
   aas:hasName “Hoekprofiel 01:Hoekprofiel 01:7576690”^^xsd:string ;
   aas:hasSTD “21.13”^^xsd:number ;
   aas:hasType “Hoekprofiel 01:Hoekprofiel 01:7576688”^^xsd:string ;
   aas:isExternal true ;
   aas:loadBearing false ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .

aas:IfcWall002 a ifc:IfcWall ;
   aas:hasBuilding “”^^xsd:string ;
   aas:hasGuid “1MlDW4hyz7thdXm8zl0rqQ”^^xsd:unicode ;
   aas:hasLocation “04 vierde verdieping”^^xsd:string ;
   aas:hasName “Hoekprofiel 1:Hoekprofiel 01:7576790”^^xsd:string ;
   aas:hasSTD “21.13”^^xsd:number ;
   aas:hasType “Hoekprofiel 1:Hoekprofiel 01:7576788”^^xsd:string ;
   aas:isExternal true ;
   aas:loadBearing false ;
   aas:project “Parkeergarage Schiphol P3”^^xsd:string ;
   aas:requires “None”^^xsd:string .
………………………
```

Partial file of the model of the C-pier:

```
@prefix : <http://www.schiphol.nl/ontology#> .
@prefix aas: <http://www.schiphol.nl/ontology#> .
@prefix ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

aas:IfcCurtainWall013 aas:hasBuilding “CP”^^xsd:string ;
   aas:hasGuid “3370RJCuH10BOlsfYayKtK”^^xsd:unicode ;
   aas:hasLocation “01_eerste verdieping”^^xsd:string ;
   aas:hasName “Curtain Wall:31_FIM_vliesgevel_generiek_profielen_75x180mm:582798”^^xsd:string ;
   aas:hasSTD “ “^^xsd:number ;
   aas:hasType “Curtain Wall:31_FIM_vliesgevel_generiek_profielen_75x180mm:628”^^xsd:string ;
   aas:isExternal true ;
   aas:project “BIM model C-Pier”^^xsd:string ;
   aas:requires “None”^^xsd:string .

aas:IfcCurtainWall014 aas:hasBuilding “CP”^^xsd:string ;
   aas:hasGuid “2iJdAhBjrDvvovprYfdF$f”^^xsd:unicode ;
   aas:hasLocation “01_eerste verdieping”^^xsd:string ;
   aas:hasName “Curtain Wall:31_FIM_vliesgevel_generiek_profielen_75x180mm:605364”^^xsd:string ;
   aas:hasSTD “ “^^xsd:number ;
   aas:hasType “Curtain Wall:31_FIM_vliesgevel_generiek_profielen_75x180mm:628”^^xsd:string ;
   aas:isExternal true ;
   aas:project “BIM model C-Pier”^^xsd:string ;
   aas:requires “None”^^xsd:string .
………………………
aas:IfcDoor031 a ifc:IfcDoor ;
   aas:ClearHeight “2345.0”^^xsd:float ;
   aas:ClearWidth “1078.0”^^xsd:float ;
   aas:hasBuilding “CP”^^xsd:string ;
```

```
   aas:hasGuid "24LaATKY96ch9U6BAA5OwG"^^xsd:unicode ;
   aas:hasLocation "01_eerste verdieping"^^xsd:string ;
                                  aas:hasName         "32_FIM_binnendeur_standaard:32_FIM_binnendeur_enkel_met_
slot_1078x2345mm:823559"^^xsd:string ;
   aas:hasSTD " "^^xsd:number ;
   aas:hasType "32_FIM_binnendeur_enkel_met_slot_1078x2345mm"^^xsd:string ;
   aas:isExternal false ;
   aas:project "BIM model C-Pier"^^xsd:string ;
   aas:requires "None"^^xsd:string .

aas:IfcDoor033 a ifc:IfcDoor ;
   aas:ClearHeight "2122.5"^^xsd:float ;
   aas:ClearWidth "1149.32046113"^^xsd:float ;
   aas:hasBuilding "CP"^^xsd:string ;
   aas:hasGuid "24LaATKY96ch9U6BAA5OzE"^^xsd:unicode ;
            aas:hasName    "32_FIM_CP_vlakvulling_deur_met_glas_rond:32_FIM_CP_vlakvulling_deur_met_glas_
rond:823562"^^xsd:string ;
   aas:hasSTD " "^^xsd:number ;
   aas:hasType "32_FIM_CP_vlakvulling_deur_met_glas_rond"^^xsd:string ;
   aas:isExternal false ;
   aas:project "BIM model C-Pier"^^xsd:string ;
   aas:requires "None"^^xsd:string .
.........................
aas:IfcTransportElement774 a aas:Movingwalkway ;
   aas:hasBuilding "CP"^^xsd:string ;
   aas:hasGuid "3HODfZpovCuBGqUx14FpoK"^^xsd:unicode ;
   aas:hasLocation "00_begane grond"^^xsd:string ;
  aas:hasName "66_FIM_roltrap_generiek_35_graden:66_FIM_roltrap_generiek_35_graden:485005"^^xsd:string
;
   aas:hasSTD " "^^xsd:number ;
   aas:hasType "66_FIM_roltrap_generiek_35_graden"^^xsd:string ;
   aas:project "BIM model C-Pier"^^xsd:string ;
   aas:requires "None"^^xsd:string .

aas:IfcTransportElement775 a aas:Movingwalkway ;
   aas:hasBuilding "CP"^^xsd:string ;
   aas:hasGuid "2g_iJYbR9EoRjX2tb1UBYk"^^xsd:unicode ;
   aas:hasLocation "01_eerste verdieping"^^xsd:string ;
    aas:hasName "66_FIM_rolpaden_loopband_model_kort_generiek:66_FIM_rolpaden_loopband_model_kort_
generiek:561987"^^xsd:string ;
   aas:hasSTD " "^^xsd:number ;
   aas:hasType "66_FIM_rolpaden_loopband_model_kort_generiek"^^xsd:string ;
   aas:project "BIM model C-Pier"^^xsd:string ;
   aas:requires "None"^^xsd:string .
.........................
aas:IfcWall001 a ifc:IfcWall ;
   aas:hasBuilding "CP"^^xsd:string ;
   aas:hasGuid "0cWu96VaHBTuDp9EdsHjQ5"^^xsd:unicode ;
   aas:hasLocation "00_begane grond"^^xsd:string ;
   aas:hasName "Basic Wall:22_FIM_WA_binnnenwand_generiek_250:499176"^^xsd:string ;
   aas:hasSTD " "^^xsd:number ;
   aas:hasType "Basic Wall:22_FIM_WA_binnnenwand_generiek_250:499174"^^xsd:string ;
   aas:isExternal false ;
   aas:loadBearing false ;
   aas:project "BIM model C-Pier"^^xsd:string ;
   aas:requires "None"^^xsd:string .

aas:IfcWall002 a ifc:IfcWall ;
```

```
    aas:hasBuilding “CP”^^xsd:string ;
    aas:hasGuid “0cWu96VaHBTuDp9EdsHjSE”^^xsd:unicode ;
    aas:hasLocation “00_begane grond”^^xsd:string ;
    aas:hasName “Basic Wall:22_FIM_WA_binnnenwand_generiek_250:499177”^^xsd:string ;
    aas:hasSTD “ “^^xsd:number ;
    aas:hasType “Basic Wall:22_FIM_WA_binnnenwand_generiek_250:499174”^^xsd:string ;
    aas:isExternal false ;
    aas:loadBearing false ;
    aas:project “BIM model C-Pier”^^xsd:string ;
    aas:requires “None”^^xsd:string .
……………………..
```

Appendix VIII: Partial validation report

Validation Report
SCHIPHOL PARKEERGARAGE P3_BOUWKUNDIG.ifc
2018-02-26, 14:21

OBJECT	GLOBALID	NAME	OBJECTTYPE
http://www.schiphol.nl/ontology#IfcCurtainWall027	18nj49GpTEqxkL6Njb59e0	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:7825401	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8215902
http://www.schiphol.nl/ontology#IfcCurtainWall030	18nj49GpTEqxkL6Njb59IT	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7825764	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall045	1NyWnhZCT2ehlVzepoWI0X	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:7831997	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcCurtainWall046	1NyWnhZCT2ehlVzepoWI9C	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:7832528	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcCurtainWall059	0Tylozvbv9HOq4XkPGtXSN	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:8222964	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall063	0Lpv8ruNnBCezKnhPwZsXS	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8280179	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8215902
http://www.schiphol.nl/ontology#IfcCurtainWall064	0Lpv8ruNnBCezKnhPwZsk$	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:8280976	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall065	2DltYyHoX1qAPYNrXatpha	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:8283157	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall066	3fSXgS3dz07BGYt7m0DEcc	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Voorgevel:8284046	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Voorgevel:8215822
http://www.schiphol.nl/ontology#IfcCurtainWall067	1mcT3AlrTDaxHm_SxJ7dWC	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8289023	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8215902
http://www.schiphol.nl/ontology#IfcCurtainWall068	1mcT3AlrTDaxHm_SxJ7daj	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8289246	Curtain Wall:BN_21.1_Vliesgevel_trappenhuis_01 Zijgevel:8215902
http://www.schiphol.nl/ontology#IfcCurtainWall561	2xAfueCtHC9hmGn9187F5V	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:9351209	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcCurtainWall577	2FKFV6Uu5EoBRiGXjqVQRJ	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:9404724	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall578	2FKFV6Uu5EoBRiGXjqVQRT	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:9404730	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 01:7009822
http://www.schiphol.nl/ontology#IfcCurtainWall582	3G3Miq5kH4lgjMZYUBVF_k	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:9421549	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcCurtainWall583	3G3Miq5kH4lgjMZYUBVF_m	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:9421555	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcCurtainWall584	3G3Miq5kH4lgjMZYUBVF_w	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:9421561	Curtain Wall:BN_21.1_Vliesgevel_Achtergevel 02:8223256
http://www.schiphol.nl/ontology#IfcDoor011	0ZLDNXVMD4$geq9hXI8L7c	BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7703961	BN_OPG_Schuifdeur
http://www.schiphol.nl/ontology#IfcDoor012	0ZLDNXVMD4$geq9hXI8L7d	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7703962	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor017	0i0LAIlBX4qwVzNb4dR8IS	BN_OPG_Schuifdeur:BN_Schuifdeur Laag 0:7742246	BN_Schuifdeur Laag 0
http://www.schiphol.nl/ontology#IfcDoor018	0i0LAIlBX4qwVzNb4dR8IT	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7742247	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor019	0PczFb92j4pRTXUkG2Kig9	BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7778705	BN_OPG_Schuifdeur
http://www.schiphol.nl/ontology#IfcDoor020	0PczFb92j4pRTXUkG2KigA	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7778706	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor021	0PczFb92j4pRTXUkG2KigF	BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7778711	BN_OPG_Schuifdeur
http://www.schiphol.nl/ontology#IfcDoor022	0PczFb92j4pRTXUkG2Kig0	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7778712	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor023	0PczFb92j4pRTXUkG2Kig5	BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7778717	BN_OPG_Schuifdeur
http://www.schiphol.nl/ontology#IfcDoor024	0PczFb92j4pRTXUkG2Kig6	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7778718	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor025	0PczFb92j4pRTXUkG2Kigx	BN_OPG_Schuifdeur:BN_OPG_Schuifdeur:7778723	BN_OPG_Schuifdeur
http://www.schiphol.nl/ontology#IfcDoor026	0PczFb92j4pRTXUkG2Kigy	BN_Deur met zijlicht_01:BN_Deur met zijlicht 01:7778724	BN_Deur met zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor028	18nj49GpTEqxkL6Njb59L_	BN_OPG_Schuifdeur:BN_Schuifdeur Laag 0 2:7825543	BN_Schuifdeur Laag 0 2
http://www.schiphol.nl/ontology#IfcDoor029	18nj49GpTEqxkL6Njb59KQ	BN_Deur met zijlicht_01:BN_Deur met zijlicht 02:7825635	BN_Deur met zijlicht 02
http://www.schiphol.nl/ontology#IfcDoor032	18nj49GpTEqxkL6Njb592L	BN_Liftdeur:BN_Liftdeur:7826796	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor033	18nj49GpTEqxkL6Njb592K	BN_Liftdeur:BN_Liftdeur:7826797	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor034	18nj49GpTEqxkL6Njb592N	BN_Liftdeur:BN_Liftdeur:7826798	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor035	18nj49GpTEqxkL6Njb592M	BN_Liftdeur:BN_Liftdeur:7826799	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor036	18nj49GpTEqxkL6Njb5929	BN_Liftdeur:BN_Liftdeur:7826800	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor037	18nj49GpTEqxkL6Njb5928	BN_Liftdeur:BN_Liftdeur:7826801	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor038	18nj49GpTEqxkL6Njb592B	BN_Liftdeur:BN_Liftdeur:7826802	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor039	18nj49GpTEqxkL6Njb592A	BN_Liftdeur:BN_Liftdeur:7826803	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor040	18nj49GpTEqxkL6Njb592D	BN_Liftdeur:BN_Liftdeur:7826804	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor041	18nj49GpTEqxkL6Njb592C	BN_Liftdeur:BN_Liftdeur:7826805	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor042	18nj49GpTEqxkL6Njb592F	BN_Liftdeur:BN_Liftdeur:7826806	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor043	18nj49GpTEqxkL6Njb592E	BN_Liftdeur:BN_Liftdeur:7826807	BN_Liftdeur
http://www.schiphol.nl/ontology#IfcDoor047	0vDq0dU9L1ux88a0AW7Ou5	BN_Schachtdeur_01:BN_Schachtdeur_01:7935638	BN_Schachtdeur_01
http://www.schiphol.nl/ontology#IfcDoor048	0vvl_kDsH1dwVu4r93jb4Z	NL_32_binnendeur:Werkkast 1000x2300 Hout:8135692	Werkkast 1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor049	0vvl_kDsH1dwVu4r93jbCt	NL_32_binnendeur:Werkkast 1000x2300 Hout:8136216	Werkkast 1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor050	0vvl_kDsH1dwVu4r93jbE1	NL_32_binnendeur:Werkkast 1000x2300 Hout:8136366	Werkkast 1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor051	3mMtJmQxz0zxNZu_VIJ$hu	NL_32_binnendeur:Werkkast 1000x2300 Hout:8141164	Werkkast 1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor052	3mMtJmQxz0zxNZu_VIJ$hg	BN_Schachtdeur_01:BN_Schachtdeur_01:8141182	BN_Schachtdeur_01
http://www.schiphol.nl/ontology#IfcDoor070	2e8OQ1VrHEmxLT7IMJodpS	BN_Zijlicht_01:BN_Zijlicht 01:8431273	BN_Zijlicht 01
http://www.schiphol.nl/ontology#IfcDoor071	04VcqzDSHET8Wocd16G1A0	BN_Bedieningskast:BN_Bedieningskast:8436718	BN_Bedieningskast
http://www.schiphol.nl/ontology#IfcDoor072	04VcqzDSHET8Wocd16G1PK	BN_Bedieningskast:BN_Bedieningskast:8437562	BN_Bedieningskast
http://www.schiphol.nl/ontology#IfcDoor073	1RoUUNDWTDzveVZjS6TI75	NL_32_binnendeur:Werkkast 1000x2300 Hout:8440955	Werkkast 1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor077	1xe1yad6H8Ox9ehwaME1s0	BN_Buitendeur Dubbel_Hout_01:2200x2300 Hout:8513218	2200x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor078	09LYaY8Uv9hvaFpiv8Por4	BN_Deur_JAZO_01:Traforuimte:8532381	Traforuimte
http://www.schiphol.nl/ontology#IfcDoor255	3P94bT3_XF3PAlou4MV42d	BN_Rooster_JAZO_01:02:8611852	02
http://www.schiphol.nl/ontology#IfcDoor256	3P94bT3_XF3PAlou4MV48P	BN_Rooster_JAZO_01:dm = 1100mm:8612530	dm = 1100mm
http://www.schiphol.nl/ontology#IfcDoor431	0eYe_c3d5BkBLbs$bBUqXZ	NL_32_binnendeur:1000x2300 Hout:9074231	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor579	2FKFV6Uu5EoBRiGXjqVQ_1	BN_Betondorpel_01:Betondorpel 225x150mm:9406566	Betondorpel 225x150mm
http://www.schiphol.nl/ontology#IfcDoor580	2FKFV6Uu5EoBRiGXjqVQwO	BN_Betondorpel_01:Betondorpel 225x150mm:9406847	Betondorpel 225x150mm
http://www.schiphol.nl/ontology#IfcDoor581	2FKFV6Uu5EoBRiGXjqVQud	BN_Betondorpel_01:Betondorpel 270x150mm:9406912	Betondorpel 270x150mm
http://www.schiphol.nl/ontology#IfcDoor585	3QSgcJ9v17shAVWK86x0by	BN_Betondorpel_01:Betondorpel 225x150mm:9422295	Betondorpel 225x150mm
http://www.schiphol.nl/ontology#IfcDoor586	3QSgcJ9v17shAVWK86x0bp	BN_Betondorpel_01:Betondorpel 270x150mm:9422296	Betondorpel 270x150mm
http://www.schiphol.nl/ontology#IfcDoor649	0bhfBEJTjBGRvkuG4OUudg	BN_Buitendeur Hout_01:1100x2300 Hout:9667951	1100x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor650	0bhfBEJTjBGRvkuG4OUudr	BN_Buitendeur Dubbel_Hout_01:2200x2300 Hout:9667952	2200x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor651	0bhfBEJTjBGRvkuG4OUudq	BN_Buitendeur Hout_01:1100x2300 Hout:9667953	1100x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor652	1vgqCbyEz0mQFjDNeWZenM	BN_Buitendeur Hout_01:1100x2300 Hout:9711653	1100x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor658	3gX$h2a2vBhB6n7nmKMYSu	BN_Rooster_JAZO_01:03:9843624	03
http://www.schiphol.nl/ontology#IfcDoor659	38Wc7lEmXER8wl2aEV3CcU	BN_Schachtdeur_01:BN_Schachtdeur_01:9848179	BN_Schachtdeur_01
http://www.schiphol.nl/ontology#IfcDoor660	38Wc7lEmXER8wl2aEV3CcM	BN_Schachtdeur_01:BN_Schachtdeur_01:9848187	BN_Schachtdeur_01
http://www.schiphol.nl/ontology#IfcDoor661	38Wc7lEmXER8wl2aEV3CbX	BN_Schachtdeur_01:BN_Schachtdeur_01:9848204	BN_Schachtdeur_01
http://www.schiphol.nl/ontology#IfcDoor663	3poBVCH6PFkAhn1PVgVxzH	NL_32_binnendeur:1000x2300 Hout:11003431	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor664	3poBVCH6PFkAhn1PVgVx_m	NL_32_binnendeur:1000x2300 Hout:11003590	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor665	3poBVCH6PFkAhn1PVgVx6u	NL_32_binnendeur:1000x2300 Hout:11004110	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor666	3poBVCH6PFkAhn1PVgVx0V	NL_32_binnendeur:1000x2300 Hout met dorpel 2:11004265	1000x2300 Hout met dorpel 2
http://www.schiphol.nl/ontology#IfcDoor667	3zPiJi0A90qQ5kJoVAvcjB	NL_32_binnendeur:1000x2300 Hout met dorpel:11007816	1000x2300 Hout met dorpel
http://www.schiphol.nl/ontology#IfcDoor668	3PZn0H_OLE3OeDwxh84Lw0	NL_32_binnendeur:1000x2300 Hout met dorpel:11010342	1000x2300 Hout met dorpel
http://www.schiphol.nl/ontology#IfcDoor669	3PZn0H_OLE3OeDwxh84LWy	NL_32_binnendeur:1000x2300 Hout:11011994	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor670	3PZn0H_OLE3OeDwxh84LXN	NL_32_binnendeur:1000x2300 Hout:11012081	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor671	3PZn0H_OLE3OeDwxh84LUN	NL_32_binnendeur:1000x2300 Hout:11012145	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor672	3PZn0H_OLE3OeDwxh84LVU	NL_32_binnendeur:1000x2300 Hout:11012216	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor673	3PZn0H_OLE3OeDwxh84LTW	NL_32_binnendeur:1000x2300 Hout:11012294	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor674	3PZn0H_OLE3OeDwxh84LTT	NL_32_binnendeur:1000x2300 Hout met dorpel 2:11012347	1000x2300 Hout met dorpel 2
http://www.schiphol.nl/ontology#IfcDoor675	3PZn0H_OLE3OeDwxh84KrR	NL_32_binnendeur:1000x2300 Hout:11014909	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor676	3PZn0H_OLE3OeDwxh84KoR	NL_32_binnendeur:1000x2300 Hout:11014973	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor677	3PZn0H_OLE3OeDwxh84KmY	NL_32_binnendeur:1000x2300 Hout:11015044	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor678	3PZn0H_OLE3OeDwxh84Kk1	NL_32_binnendeur:1000x2300 Hout:11015207	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor679	3PZn0H_OLE3OeDwxh84Kl5	NL_32_binnendeur:1000x2300 Hout met dorpel 2:11015267	1000x2300 Hout met dorpel 2
http://www.schiphol.nl/ontology#IfcDoor690	0pODNkk9544h1CDUBBRx5B	NL_32_binnendeur:1000x2300 Hout:11016855	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor691	0pODNkk9544h1CDUBBRxSR	NL_32_binnendeur:1000x2300 Hout:11017415	1000x2300 Hout
http://www.schiphol.nl/ontology#IfcDoor692	3kgvF1mqX1SfZsPegsE5$F	NL_32_binnendeur:1000x2300 Hout met dorpel:11036089	1000x2300 Hout met dorpel
http://www.schiphol.nl/ontology#IfcWall001	1MIDW4hyz7thdXm8zl0rs_	Hoekprofiel 01:Hoekprofiel 01:7576690	Hoekprofiel 01:Hoekprofiel 01:7576688
http://www.schiphol.nl/ontology#IfcWall002	1MIDW4hyz7thdXm8zl0rqQ	Hoekprofiel 1:Hoekprofiel 01:7576790	Hoekprofiel 1:Hoekprofiel 01:7576788
http://www.schiphol.nl/ontology#IfcWall003	1MIDW4hyz7thdXm8zl0rzD	Hoekprofiel 2:Hoekprofiel 01:7577217	Hoekprofiel 2:Hoekprofiel 01:7577215
http://www.schiphol.nl/ontology#IfcWall004	1MIDW4hyz7thdXm8zl0rz1	Hoekprofiel 3:Hoekprofiel 01:7577229	Hoekprofiel 3:Hoekprofiel 01:7577227
http://www.schiphol.nl/ontology#IfcWall005	1MIDW4hyz7thdXm8zl0sSn	Gevelprofiel 7:Gevelprofiel 01:7579389	Gevelprofiel 7:Gevelprofiel 01:7579387
http://www.schiphol.nl/ontology#IfcWall006	1MIDW4hyz7thdXm8zl0sR7	Gevelprofiel 1:Gevelprofiel 01:7579403	Gevelprofiel 1:Gevelprofiel 01:7579401
http://www.schiphol.nl/ontology#IfcWall007	1MIDW4hyz7thdXm8zl0sQU	Gevelprofiel 2:Gevelprofiel 01:7579474	Gevelprofiel 2:Gevelprofiel 01:7579472
http://www.schiphol.nl/ontology#IfcWall008	1MIDW4hyz7thdXm8zl0sQy	Gevelprofiel 3:Gevelprofiel 01:7579504	Gevelprofiel 3:Gevelprofiel 01:7579502
http://www.schiphol.nl/ontology#IfcWall009	1MIDW4hyz7thdXm8zl0sP2	Gevelprofiel 4:Gevelprofiel 01:7579534	Gevelprofiel 4:Gevelprofiel 01:7579532
http://www.schiphol.nl/ontology#IfcWall010	2HPWTa_yvE4BkqbuBcGxu3	Gevelprofiel 6:Gevelprofiel 01:7579654	Gevelprofiel 6:Gevelprofiel 01:7579652
http://www.schiphol.nl/ontology#IfcWall013	37e_V0_YfBYPBPsCG9CzAd	BN_Kader Inrit:BN_Kader Inrit:7726089	BN_Kader Inrit:BN_Kader Inrit:8657973

STD	PROJECT	BUILDING	STOREY	SPACE	CLEARWIDTH	CLEARHEIGHT	CLEARDEPTH
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	2506.0	2385.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2200.0	2350.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2599.5	2395.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	2300.0	2430.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1480.0	2427.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	1030.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	1030.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	1030.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	1030.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	1480.0	2427.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1238.5	2395.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	545.0	2410.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	545.0	2410.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	1030.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2350.0	2375.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1921.34999895	2387.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2010.0	416.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	760.0	2377.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	3153.5171004	150.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	4514.4828996	150.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	4182.0	150.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	4410.5085502	150.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	4151.4914498	150.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1250.0	2375.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	2350.0	2375.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1250.0	2375.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1250.0	2375.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1200.0	1074.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	1480.0	2427.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	1480.0	2427.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	1480.0	2427.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	1114.0	2367.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	1114.0	2369.0	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	04 vierde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	01 eerste verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	02 tweede verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	03 derde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	05 vijfde verdieping	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	Parkeergarage Schiphol P3	[redacted]	00 begane grond	[redacted]	[redacted]	[redacted]	[redacted]